Jet Propulsion Laboratory, Faculty of Aerospace, Technion-IIT

Dynamic Scaling of Prototypes Using Radius of Gyration Method:

Theory, Laboratory and Flight Tests

I - Report for Period: Aug. 1 to Sep. 30, 1991.

[Attention: Capt. J. Wigle, USAF, AFOSR, EOARD, UK]

II - Progress Report for Period: Oct. 1 Dec. 20,1991.

[Attention: Dr. D. W. Repperger, AL/WPAFB,

via USAF/AFOSR/EOARD, U.K., Dr. W. Calarese.]

Benjamin Gal-Or

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1991	Final Report, 1 Aug - 20 Dec 1991

4. TITLE AND SUBTITLE

Dynamic Scaling of Prototypes Using Radius of Gyration Method: Theory, Laboratory and Flight Tests

5. FUNDING NUMBERS

SPC-91-4003

6. AUTHOR(S)

Benjamin Gal-Or

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Jet Propulsion Laboratory
Department of Aerospace Engineering
Technion - Israel Institute of Technology
Technion City, Haifa 32000, Israel

8. PERFORMING ORGANIZATION REPORT NUMBER

SPC-91-4003

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
PSC 802 BOX 14
FPO AE 09499-0200

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

SPC-91-4003

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The potential introduction of new Post-Stall (PST), air-to-air and air-to-ground maneuvers and tactics via thrust-vectoring control (TVC) may dramatically increase fighter-aircraft performance at low subsonic speeds. TVC is now expected to double or triple maximum nose turning rates, especially under non-conventional, rapid, rigid-body-type-rotations/translations, or during PST-up/down whippings. It is also expected to expand flight envelopes beyond conventional limits, reduce signatures, improve STOL capabilities and enhance safety margins, especially at takeoff and landing and in the prevention or recovery from any spin situation. This interim Progress Report deals with the generation of such data for future USAF PST-TVF-15S.

14. SUBJECT TERMS	15. NUMBER OF PAGES
	141
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19, SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

Jet Propulsion Laboratory, Faculty of Aerospace, Technion-IIT

Dynamic Scaling of Prototypes Using Radius of Gyration Method:

Theory, Laboratory and Flight Tests

I - Report for Period: Aug. 1 to Sep. 30, 1991.

[Attention: Capt. J. Wigle, USAF, AFOSR, EOARD, UK]

USAF Special Project SPC-91-4003; Technion Res. No.: 160-662

II - Progress Report for Period: Oct. 1 Dec. 20,1991.

[Attention: Dr. D. W. Repperger, AL/WPAFB,

via USAF/AFOSR/EOARD, U.K., Dr. W. Calarese.]

USAF Special Project SPC-91-4003; Technion Res. No.: 160-662

Principal Investigator: Dr. Benjamin Gal-Or

Prof. & Head, Jet Propulsion Laboratory,

Faculty of Aerospace, Technion-Israel Institute of Technology, Haifa,

FAX: 972-4-221581, Tel. 972-4-348066, 292807, 292435.

Copies: **Col. J. Tedor,** Deputy Director, Human Systems Division, BAFB,

and **Capt. D. D. Baumann** and **Mr. D. Bowers,** FDD, WL, WPAFB.

Table of Contents

* * *

Outline

The potential introduction of new Post-Stall [PST], air-to-air and air-to-ground maneuvers and tactics via thrust-vectoring-control [TVC] may dramatically increase fighter-aircraft performance at low subsonic speeds. TVC is now expected to double or triple maximum nose turning rates, especially under non-conventional, rapid, rigid-body-type-rotations/translations, or during 'PST-up/down whippings'. It is also expected to expand flight envelopes beyond conventional limits, reduce signatures, improve STOL capabilities and enhance safety margins, especially at takeoff and landing and in the prevention or recovery from any spin situation [see **Fundamental Concepts of Vectored Aircraft**' at Report end].

Yet, TVC introduces unknown human-aircraft domains which may or may not limit combat effectiveness due to pilot tolerances during such supermaneuvers.

This laboratory conducted the first PST-TVC flight tests of unmanned vehicles [1987] and the first 'Cobra' maneuvers by TVC [1989]. It is also the only laboratory that uses powered, dynamically-scaled PST-TV RPV models of USAF fighters to investigate new maneuvers, advanced yaw-pitch or roll-yaw-pitch TV-nozzles, and PST-TV-agility limitations, including the generation of 'PST-TV-induced-g-loads' data for USAF centrifuge emulation of pilot's motion fields under such flight conditions. Currently there is no other source for such data.

This interim Progress Report deals with the generation of such data for future USAF PST-TV F-15s.

PST-TV F-22 and F-16 RPVs may be included later, i.e., beyond this contract. The F-22 project is currently independently funded by USAF/AFOSR/EOARD. A related study on F-22-type yaw-pitch TV-nozzles is currently funded by PWA. The last one includes TV-nozzle tests with our hot-propulsion jet engine rigs/facilities.

Reported below are recent improvements in the performance of our recently-modified PST-TV F-15 RPV, including upgrading of its PST-TV-pitch capability, adding accelerometers probes, and improving our flight recording and ground computers, flight-testing procedures, post-flight testing analysis methodology and software/hardware, as well as a few preliminary results from a flight test conducted on Nov. 7, 91. Also included are a few theoretical assertions and rules related to this field.

While we constantly employ only the best flyers in this country, the new complex systems/maneuvers/software/probes have claimed a high price: Six crashes, about two dozen hard landings and numerous aborted takeoffs. The sources of these difficulties are enumerated below in '**Technology Limits**'.

The preliminary flight test results reported here provide a proof that a highly improved performance has been reached by the recently upgraded PST-TV F-15 RPV: up to about **200** deg/sec pitch rate [and up to about 110 deg pitch angle at the reversal point of the SACOM] during pseudo 'Cobra' and Herbst maneuvers.

Using our Dynamic Scale Factors [DSF], these rates translate to **$[200][7]^{-0.5}$ = 75 deg/sec pitch rate** for the full-scale F-15s at about 0.3-0.4M, [and about 110 degrees pitch angle at the reversal point of the SACOM]. This value is **about three times the maximum current pitch rate ['corner turn rate'] of the F-15As.** [Note: Up to about **0.3 sec delay time** has been measured in the laboratory in deflecting the jet vane of the flying model to full 20 degrees. Hence, the net pitch rate can be further increased by reducing this delay time. Higher pitch rates are also extractable by changing the TV-nozzle limits so that very rapid 30 degrees deflections become possible. Changing the static stability margin may also affect this value.]

Consequently, the enhanced performance PST-TV F-15 RPV [which had been designed and developed from Aug. 1 to Sep. 30, 91], allows to perform new PST-TV maneuvers as well as pseudo 'Cobra' and 'Herbst' maneuvers, in line with the Tasks of this research [see below].

Video tape No 7 shows these rapid pseudo 'Cobra' and Herbst SACOM-maneuvers.

Following editing and additional flight tests, we shall formally submit 2 copies of this video tape, one to **WL/AL/WPAFB**, the other to the **Human Systems Division at BAFB**.

Research Objectives

The research objectives of the integrated USAF/AFOSR/EOARD contracts described below are defined by the following tasks and statements of work:

By employing dynamically-scaled, 1/7-scale, PST-TV F-15 RPVs which have been designed, constructed, instrumented, calibrated and flight tested via

(i) - USAF/AFOSR/EOARD Grant Number AFOSR 89-0445 [April 1, 1989 - June 30, 1991],

(ii) - [from Aug. 1 to Sep. 30, 91] USAF/AFOSR/EOARD Special Project SPC-91-4003 [EOARD], and

(iii) - USAF/ AL-WPAFB and Human Systems Division, BAFB, Special Project SPC-91-4003 [via EOARD],

perform the following Tasks:

Task 1. Add accelerometers and noise filters and improve the thrust-vectoring moments and actuation systems to extract maximum possible 'g-onsets'. Gradually adapt/upgrade instrumentation, onboard and post-flight-analysis computers to the required expanded performance envelopes.

Task 2. Measure G_x, G_y, G_z histories 20 times per second via the onboard computer-recording at the pilot's station throughout the PST-TV maneuvers. Video tape the maneuvers with G_x, G_y, G_z histories simultaneously superimposed on the tape. Simultaneously measure velocity, angle-of-attack, sideslip angle and yaw, roll and pitch angular velocities, and accelerations, as well as flyer's conventional, TV, or TV + conventional command histories, 40 times per second for each variable.

Task 3. Establish new, expanded PST-TV envelopes which are of interest to both the BAFB [Ref. Col. J. Tedor], and to WPAFB [Ref Dr. D. W. Repperger], including negative and positive "Cobra" and "Herbst" supermaneuvers.

Task 4. Gradually design and conduct Standard Agility Comparison Maneuvers [SACOM], so as to establish PST-TV agility limitations by pilot tolerances

Task 5. Produce meaningful test data that are useful to design human-PST-TV agility limiters, new centrifuges, etc.

Task 6. Report the results and conclusions from tasks 1-5 via a few intermediate Reports.

Task 7. Document these results in a technical report supplemented with computer/instrumentation software, including significant raw flight-test data extracted from the onboard computer and proper graphs of $\mathbf{G_x}$, $\mathbf{G_y}$, $\mathbf{G_z}$ envelopes. Submit this Report , Via USAF/AFOSR/EOARD, no later than Aug. 31, 1992.

To perform the aforementioned tasks we had to improve the PST-TV capability of our F-15 flying model. That capability was hindered by a few 'Technology Limits' [see below].

These limitations centered around the limited agility provided by External Thrust Vectoring [ETV] in comparison with Internal Thrust Vectoring [ITV].

Consequently our main tasks for the period Aug. 1 to Sep 30, 91 were to design, construct, calibrate and flight test an expanded PST-TV-capability F-15 flying model. To conform with WL/AL needs, that capability is first centered around maximization of pitch PST-TV-agility, in particular the maximum agility extractable from new and modified 'Cobra' and Herbst maneuvers.

Therefore, a pitch-only ITV plate was designed for this particular purpose **[Fig. A]**. As depicted, this nozzle retains the [higher thrust-to-weight ratio] circular nozzle

shape, therby retaining the highest potentials to maximize pitch PST-TV agility.

Its first flight test was conducted on Nov. 7, 91. However, while the Herbst maneuver was conducted, the 'Cobra' one has not yet been well performed. What has been demonstrated may be termed 'rapid up/down whipping' during pseudo horizontal PST-TV maneuvers [see theoretical definitions below and **Video Tape No. 7**].

To minimize the dangers of instrumentation failures during flight tests, as happened on Nov. 7, and to increase the reliability and efficiency of the flight tests, we have, most recently:

1 - upgraded/improved our **Flight Recording Onboard and Ground Computers**. Since the changes involve hardware and ROM, we next refer to these computers as the 3rd-generation flight computers. That change has been performed in mid Dec. 91 by the PCSI company of Haifa. Drawings and modified software lists are provided below for a potential future USAF use and/or reconstruction effort to duplicate this work. This applies also to the other works listed below.

2 - purchased a fast [32 MHz via L.M., 5MB internal memory, 60Mb HD with 12MS time delays and suitable connections to car's battery] **386-based notebook computer**,

3 - developed a **new computer-software** for extracting much faster computer-to-computer data transfer [<u>2.5 minutes instead of 45 minutes</u>],

4 - developed a **modified PROCOMM** program for substantiating job No. 3,

5 - designed, constructed and calibrated new accelerometers to measure $\mathbf{G_x}$, $\mathbf{G_y}$, $\mathbf{G_z}$ envelopes/histories during the maneuvers,

6 - designed, constructed and calibrated new **noise filters** to eliminate engine noise from the measurement of $\mathbf{G_x}$, $\mathbf{G_y}$, $\mathbf{G_z}$ envelopes/histories.

7 - trained the team to master the newly-instrumented/upgraded PST-TV F-15 RPVs and the new post-flight procedures, calibration equations, chart generation, etc.

Tentative Theoretical Assertions

Fundamental studies conducted during the period of this report [Cf. Report end] indicate the following tentative rules/assertions for positive 'Cobra-Type-SACOM' **[Cf. p. 10, 11, 13]** in comparison with conventional SACOM attempting to acquire the same target:

(i) - Crossing into negative-**g** domains depends on **g**-reversing-time/position, C_L, **C_D-AoA**, **V**, altitude, pilot's distance from **CR**, & sign and duration of δ_V .

(ii) - The higher the [low subsonic] speed, the longer the delay time into negative-g;

(iii) - Contrary to high positive G_z-loads which characterize conventional pitch-up maneuvers [upper chart in p. 11], the faster the PST-TV-nose-turning rates towards the same target [lower chart in p. 11], or the shorter the [pitch] 'time-to-target-recover-PST-TV-maneuver', the more effective and safer it becomes, viz., for a pilot situated 'close-to-CR' to acquire the same target the following **rule** applies [Cf. p. 10, 11, 13, **(iv)** below & Appendices E and F].

[Maximum Required PST-TV-G_z-loads] <

[Maximum Required Conventional-G_z-loads] [Cf. iv below !!!]

(iv) - Adding tangential and 'centrifugal' accelerations on a pilot situated ahead of 'CR', does not change this rule/assertion, even for the fastest pitch-up/down at $M < 0.35$ PST-TV maneuvers[see p. 10 & **Appendix E by Dr. Valery Sherbaum**];

(v) - Maximum pitch-agility is affected by airframe/engine structural 'g-limitations' at high subsonic speeds, and is hardly, if at all, influenced by pilot tolerances at low

speeds.

To verify these assertions, improve TVC designs, and study PST-TV agility and tactics, the maximized G_z, G_x, G_y, **q**, **p**, **r**, α, β, and **V** envelopes are to be simultaneously measured by our dynamically-scaled PST-TV-models during newly-defined SACOMs.

Previous Works, Publications and Reports

Previous individual contributions and collective works in this field have been most recently reviewed in the following references:

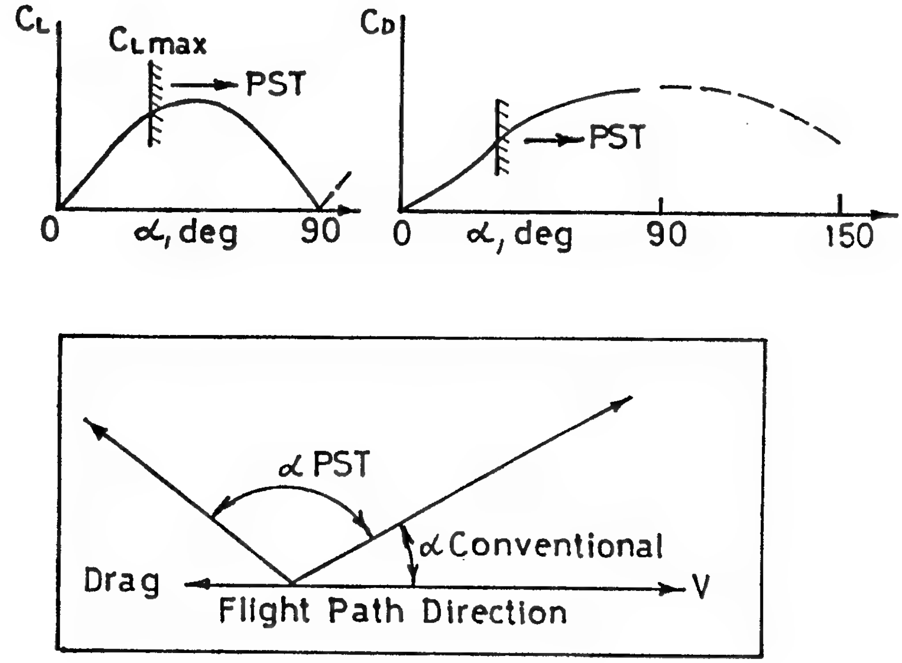
1 - Gal-Or, B., **Vectored Propulsion, Supermaneuverability and Robot Aircraft,** Springer Verlag, N.Y.- Heidelberg, 1990. 1991.

2 - Ibid., **Fundamental Concepts of Vectored Propulsion.** [AIAA] J. Propulsion, Vol. 6, No. 6, Nov.-Dec., pp. 747-757, 1990.

3 - Ibid., **Maximizing Post-Stall, Thrust-Vectoring Agility and Control Power.** [AIAA] J. Aircraft, in press.

4 - Ibid., **Tailless Vectored Fighters.** Flight Dynamics Directorate, WPAFB, USAF/AFOSR 89-0445, July 15, 1991.

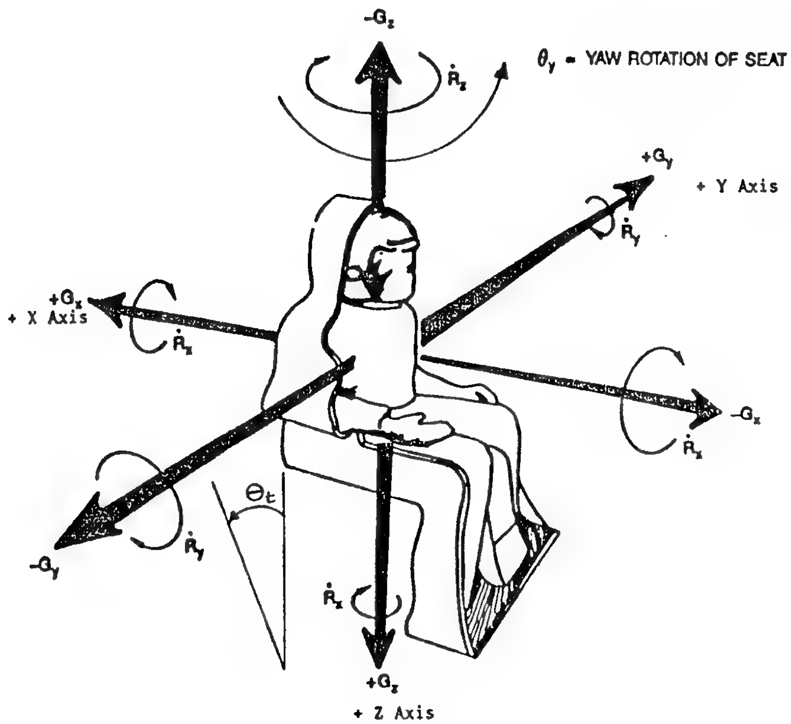
5 - Gal-Or, B. and D. D. Baumann, **Fundamental Concepts of Vectored Aircraft.** A manuscript draft is attached at Report end.



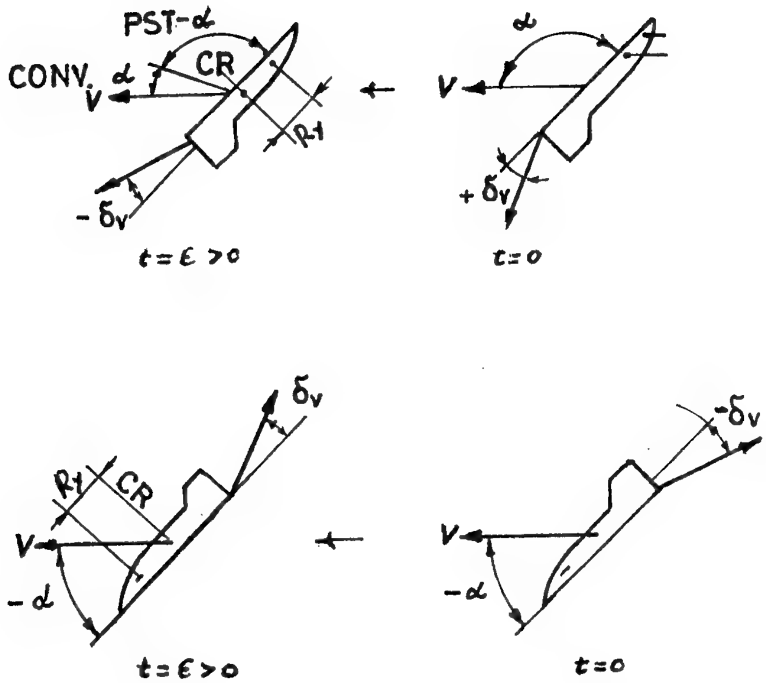
The definition of PST domain.

Note : At AoA = 90 degrees the lift vanishes, drag is maximized and roll becomes PSM.

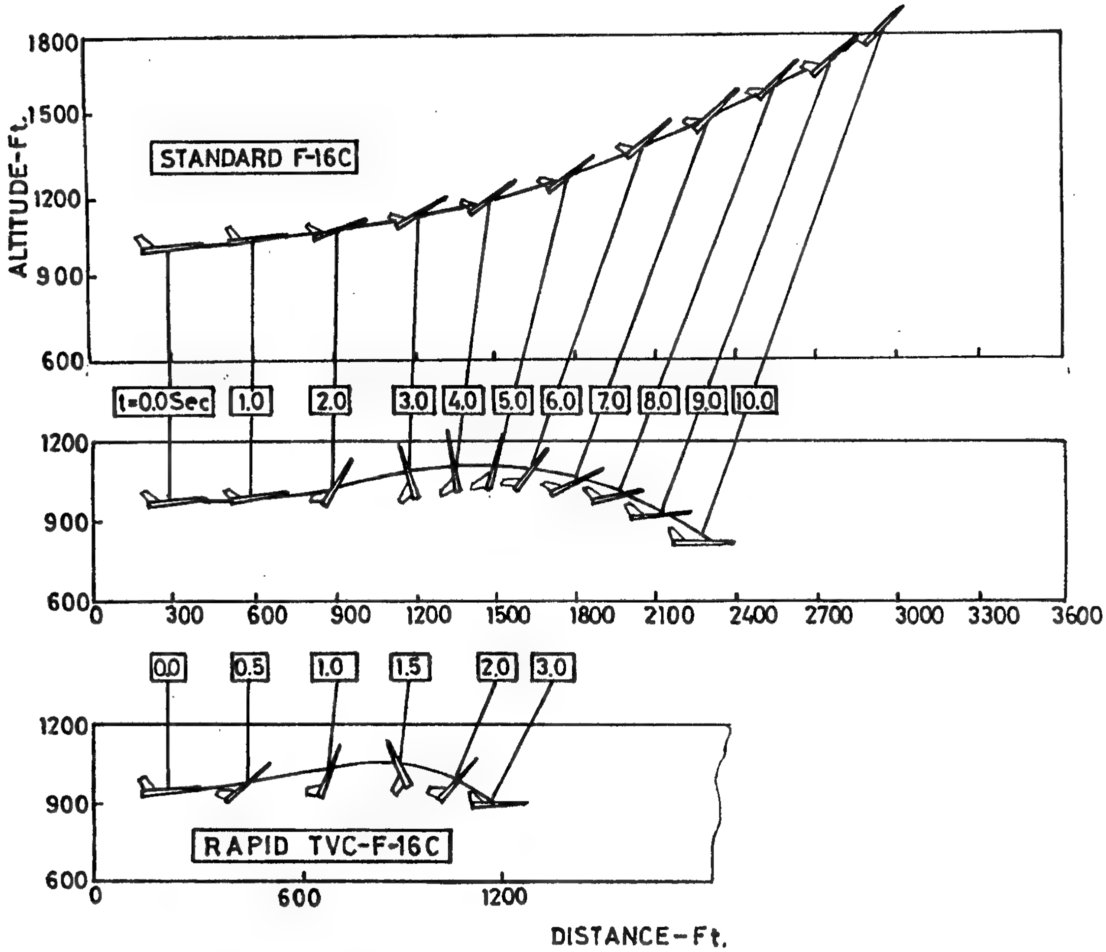
THE CONVENTIONAL DEFINITIONS OF
ACCELERATION FORCE VECTORS



THE BODY CENTERED COORDINATE SYSTEM



Positive and negative "Cobra" supermaneuvers. Effectiveness rule No. 1 requires TVC-rates not to lag behind rotational rates of conventional control, e.g., elevators, rudders and ailerons. Rule No.2 requires maximization of TV moments and rates at the reversal of pitch, roll and yaw supermaneuvers. Pitch TVC reversal is depicted here together with the radius of gyration. The figure represents the main features which require attention during pure-pitch SACOM. CR is the center of rotation.

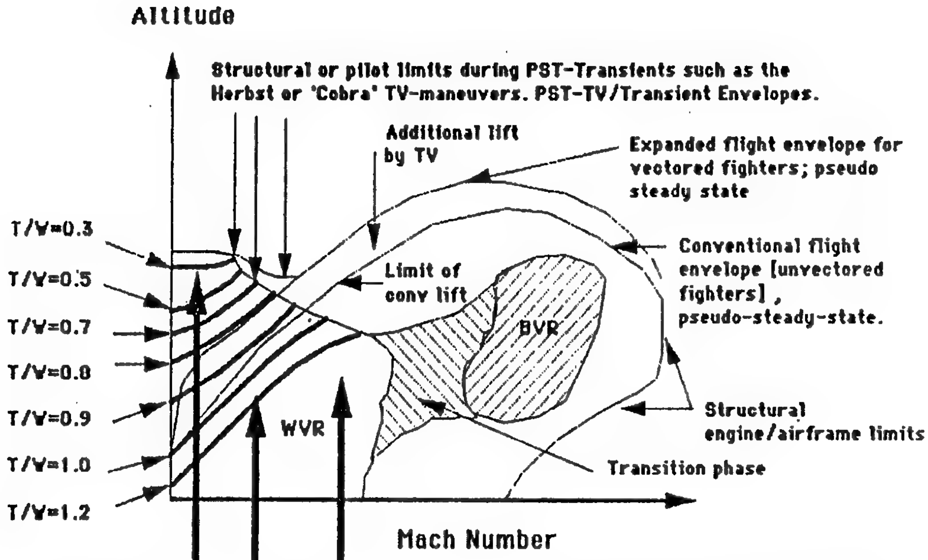


Effective TVC (lower figure) means rates which do not lag behind maximum conventional PST-rates (middle figure). Upper figure shows maximum conventional (AoA-limited) pitch-up flight control.

TV-nozzles rates must be increased from current figures (about 40 deg/sec jet-deflection rate) to beyond 400 deg/sec

TVC can "acquire-target-and-recover" at minimum time, thereby minimizing missile-flight-path/time-to target and maximizing residual speed/energy.

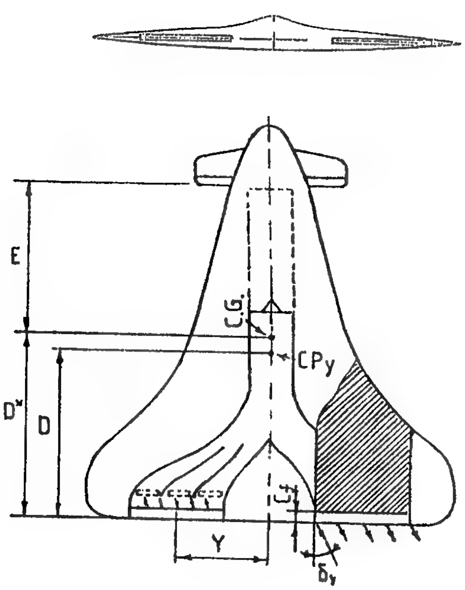
Most important : The faster-the cobra maneuver the safer it becomes to the pilot, namely, the conventional pitch-up (upper figure) generates the highest Gz loads on the pilot. (Upper 2 graphs are based on data available in the public domain. Lower graph is based on our DSF rules and flight-tests of dynamically-scaled PST-TV-F-16 and F-15 models equipped with rapidly rotating TV-jets).



Transient PST-TV Supermaneuverability Domain Where T/W < 1 or T/W > 1 and DSF Rules must be Devised to Estimate Maximal Nose/Bottom Turning Rates From Flight Tests of Correctly-Scaled Models.

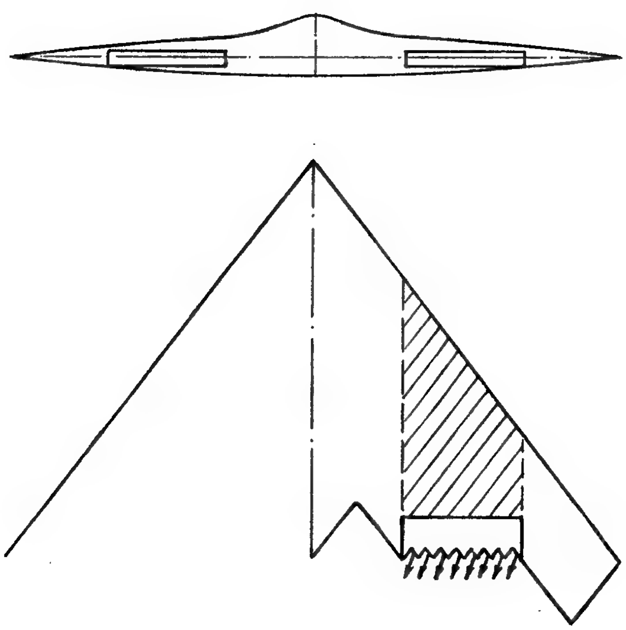
The first domain of PST-TV.

For other PST-TV-domains, including forbidden human PST-TV-domains and DSF rules, see text.



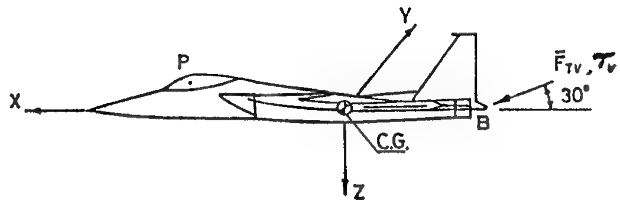
The fundamental features of Pure Vectored Aircraft (PVA)

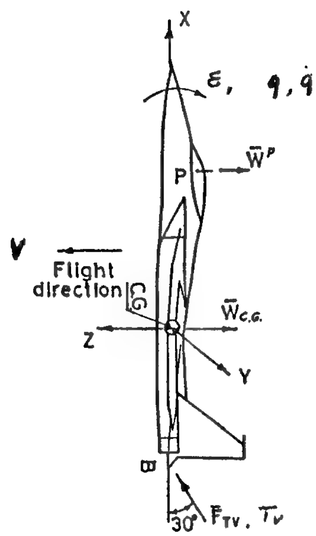
The shaded area represents super-circulation affected wing area. The propulsion system is imbedded in the fuselage and includes roll-yaw-pitch thrust-vectoring nozzles. The canard is not an essential element of PVA. The novel unmanned vehicles flight-tested in 1987 by this Laboratory have been constructed according to these features. These PVA criteria help upgrade F-15, F-16 and F-22 fighter aircraft.



PVA with reduced signatures.

Shaded area represents supercirculation-enhanced-lift area.





Definition of main coordinates and symbols. See also "Fundamental ..." at Report end.

Theory

The basic theory associated with this research has been reassessed in light of the aforementioned tasks. This preliminary effort first defines the main domains of this research from the combined point of view of Standard-Agility-Comparison-Maneuvers (SACOM) and flight-mechanics of PST-TV aircraft. It especially deals with the main relevant variables to be measured during laboratory and flight tests, as well as with expected potentials and limits involved. It is provided below and in the attached manuscript entitled **Fundamentals of Vectored Aircraft**. The man uscript

includes a section entitled Forbidden Human Space-Time Domains, which is reprinted below. A few surprising conclusions have been deduced via this preliminary effort. Comments and corrections of our mistakes are invited.

Forbidden Human Space-Time Domains

Situating an hypotethical pilot's head at CR - the 'center of rotation' [where there are no 'centrifugal' and tangential accelerations during rapid 'pure' pitch-up/down 'cobra' whippings], the normal acceleration on his head is roughly approximated by

$$G_z = \{\bar{q}s[C_L(\alpha)\cos\alpha + C_D(\alpha)\sin\alpha] + T_v\}/M\ , \qquad [26a]$$

or, for the simplifying 90-deg-AoA-pitch-SACOM-reversal [when δ_v changes sign], by

$$G_z = \{\bar{q}sC_D(90) + C_{fg}[\delta_v]\,T_j \sin\delta_v\}/M\ . \qquad [37a]$$

G_z does not change sign when δ_v does **[Fig. 9]**. The hypothetical pilot starts sensing 'negative-g' [blood flow into brain] only when, at low speed/drag values,

$$\{-T_v + \bar{q}sC_D(90)\} < 0. \qquad [37b]$$

More generally, speed reduction due to $\bar{q}s[C_L(\alpha)\cos\alpha + C_D(\alpha)\sin\alpha]$ acts to defer crossing into 'negative-g' domains, for it introduces a compensating 'positive-g' component [blood flow from brain]. Situating the pilot ahead of CR adds positive or negative tangential pitch acceleration **[Fig. 9]**, and allows simple calculations of total G_z for a realistic pilot. (cf App E)

Conclusions

(i) - Crossing into negative-g domains depends on AoA, airspeed , pilot's distance from CR, $\dot{q}$, and the value, sign and duration of the δ_v command;

(ii) - The higher the speed, the longer the delay time into negative-g domains,

(iii) - Contrary to high positive G_z-loads which characterize conventional pitch-up maneuvers [upper graph in **Fig. 10**], the faster the nose-turning rates, or the shorter

the 'time-to-target-recover-PST-TV-maneuver', the more effective, and safer, it becomes for a pilot situated 'close' to 'CR', viz., for both positive and negative pitch g-loads on the pilot,

[Maximum Possible PST-TV-G_z-loads] [Cf. lower graph in **Fig.10**] <

[Maximum PossibleConventional-G_z-loads] [Cf. upper graph in **Fig.10**]

(iv) - Adding tangential and 'centrifugal' accelerations on a pilot situated ahead of 'CR', does not change these general conclusions, even for the fastest measured PST-TV-flip-up/down at $M < 0.35$ (Cf. App. E).

(v) - Maximum pitch-agility is affected by airframe/engine structural 'g-limitations' at high subsonic speeds, and is hardly, if at all, influenced by pilot tolerances at low speeds.

To verify these conclusions, improve TVC designs, and study PST-TV agility and tactics, the maximized G_z, G_x, G_y, $\dot{q}$, $\dot{p}$, $\dot{r}$, α, β, and $\dot{V}$ envelopes are simultaneously measured by our dynamically-scaled models during very rapid pitch, roll, and sideslip SACOMs.

G_x during this SACOM includes positive [blood flow to chest] 'centrifugal' acceleration acting on the pilot from 'CR'. For this SACOM, the non-centrifugal/rotational component of G_x [when the hypothetical pilot is situated at 'CR'], is roughly approximated by

$$G_x = \{\bar{q}s[-C_L(\alpha) \sin \alpha + C_D(\alpha) \cos \alpha] - T_x\}/M \quad , \qquad [24a]$$

Similarly, the non-centrifugal/rotational portions of G_x and G_y can be measured and compared with load-approximations for **PSM** [Cf. Appendix], viz.,

$$G_x = \{-C_{fg}[\delta_y] T_i \cos \delta_y + \bar{q}sC_D[\alpha(0)]\}/M \qquad [45a]$$

$$G_y = \{-\bar{q}sC_y(\beta) - C_{fg}[\delta_y] T_i \sin \delta_y\}/M \qquad [46a]$$

The G_z, G_y, G_x pilot tolerances vary differently with the duration and rate of 'onsets'. Therefore, combined with such [a-priori known] duration/rate limitations, the

measurement envelopes may translate into forbidden human space-time agility domains for supermaneuvers.

These domains have not yet been fully explored. Their boundaries vary, inter alia, with the distance from the pilot's head to the so-called pseudo-instantaneous-center-of-rotation during different, rapid, supermaneuvers. Understanding these complex rigid-body translational, rotational, gyration, and gyroscopic phenomena, requires reassessment of a few, 'well-established', human-systems/aircraft/control/effectiveness concepts. Verification of such theoretical criteria by collecting well-defined 'flight-tested' data can therefore help the design of new centrifuge simulations [8,9] of human systems exposed to extreme PST-TV conditions, and, consequently, to establish the optimal location of the pilot's seat/head in super-agile fighters.

Radius of Gyration

The inertia tensor, $\mathbf{I_{ij}}$ (i,j = 1, 2, 3 or **x**, **y**, **z**), may be divided into an inertial tensor relative to the center of mass of the aircraft, and an inertia tensor relative to another point of reference. Hence, the quantities associated with it - principal axes, principal moments, etc. - are relative to a particular point of reference.

If the reference point is shifted from the center of mass of the aircraft to another point, as is required for improved understanding of pilot-induced rotational-agility limitations, these quantities change accordingly. The combined translational-rotational dynamics of, say, pure-pitch SACOMs, may similarly be split into two separate formulations, one purely translational and the other purely rotational about a reference point. To simplify the formulations of rigid-body dynamics and flight tests of PST-TV vehicles, one may employ the radius of gyration, which is directly related to the moments of inertia. For instance, the radius of gyration around the pitch axis of the PST-TV vehicle, $\mathbf{R_y}$, is defined by

$$\mathbf{R_y} = [\mathbf{I_y}/\mathbf{M}]^{0.5} \qquad [42]$$

where **M** is the mass of the flying vehicle [7]. Flight tests conducted by this laboratory employ the radius of gyration formulation to extract improved understanding of pilot

tolerances in dynamically-scaled-up, yaw-pitch, or roll-yaw-pitch-PST-TVC F-15, F-16 and F-22 fighter aircraft upgrades.

The measurements of G_z, G_y, G_x-envelopes - and of forbidden human space-time domains for each of these upgrades, the verification of the radius of gyration methodology for the new SACOMs, and the development of mathematical approximation methods, are sponsored by the USAF/AFOSR/EOARD, U.K. The test data are employed by the Armstrong Laboratory at WPAFB, Ohio, and by the Human System Division at BAFB, Texas. Currently there is no other source for such data [8,9].

Appendix

New Pure-Sideslip-Maneuvers With Tailless Vectored Fighters

Tailless, pure, or "ideal" thrust-vectored aircraft can perform Pure Sideslip Maneuvers [PSM] with constant [steady-state] heading, or as rapid-nose-turning transients, viz., without banking. During steady-state PSM one TV-nozzle deflects the jet in the yaw direction until its vector coincides with the side-center-of-pressure, C_{py}. This causes PSM with zero yawing-rate and banking, i.e., $\dot{r}$, $\dot{p}$, $\dot{q}$ and $\dot{\beta}$ vanish, but not β. [To perform this SACOM, the non-yawing, axial thrust generated by the 2nd TV-nozzle is reduced to equal that left-over by the 1st nozzle, so as to avoid a yawing moment on the TV-aircraft, unless transient maneuvers are required.]

During maximization of transient PSM, both nozzles are yaw-deflected in the same direction, at the fastest rate to maximum specific design-limit of δ_y values. The aim of such maneuvers with tailless configurations is to acquire the target and rapidly recover with minimal energy dissipation. [A similar PST-TV acquisition dissipates considerably more energy. Hence, to acquire any target in space-time, such a PSM-yaw is <u>a-priori</u> combined with a well-calculated roll [11]. A simplified mathematical phenomenology for assessing such advanced systems is provided next.

Consider the simplest steady-state PSM SACOM at zero AoA and zero pitch attitude with no banking and roll. During such a SACOM with pure TVC, the TV forces and moments replace the conventional ones. For a preliminary analytical assessment the conventional coupling between yaw and roll through the tail, [asymmetric-flow-over]

wing, etc., are assumed negligible for tailless pure-TVC configurations [Cf., e.g., **Figs. 3 and 4**]. Under such bold approximations the $\alpha, \dot{\alpha}, \theta, \phi, \dot{p}, \dot{q}, q, r, \delta_v, \delta_e, \delta_a, \delta_r, \delta_{le}$, $T[\Delta Z_{offset}]$, C_z, C_l, C_m, C_n terms vanish, and from eqs. 2, 6, 10 and 11 one obtains,

$$C_y \cos\beta = C_x \sin\beta \qquad [43]$$

$$\dot{V}/V = [\bar{q}s/MV][C_x \cos\beta + C_y \sin\beta] \qquad [44]$$

$$C_x = [C_{fg}[\delta_y]\, T_i \cos\delta_y] / \bar{q}s - C_D[\alpha(0)] \qquad [45]$$

$$C_y = C_y(\beta) + [C_{fg}[\delta_y]\, T_i \sin\delta_y]/\bar{q}s \qquad [46]$$

$$T_x = C_{fg}[\delta_y]\, T_i \cos\delta_y \qquad [47]$$

$$T_v = 0 \qquad [48]$$

$$T_y = C_{fg}[\delta_y]\, T_i \sin\delta_y \qquad [49]$$

δ_y incorporates two independent commands: one for each TV-nozzle [5]. Maximum possible steady-state-β-heading increases with **Y/D** values, while transient PSM-rates vary with D^* and range and rate of change of δ_y. Therefore, the fastest PSM-reversal SACOM [3] is extractable with both nozzles performing the same reversal jet-deflection, starting from zero heading, and reversing when the target has been acquired.

PSM is maximized only with high-aspect-ratio, split or s-type 2D-CD nozzles of the type depicted in **Figs. 3** and **4**. Yet, low aspect-ratio, or two axisymmetric TV-nozzles can produce reasonable PSM with tailless configurations. The combat-effectiveness of the latter is, however, limited by slow δ_y and δ_v flight-control commands, inherently shorter **Y**-moment-arms, normally, higher installed C_D values and IR/RCS signatures and the absence of $C_{[zSC]}\delta_{TV}$ and $C_{mSC}\delta_{TV}$ contributions to normal forces and moments via

$$C_z = C_{[zSC]}\delta_{TV} \qquad [12a]$$

$$C_m = C_{mSC}\delta_{TV} \qquad [14a]$$

During pitch-down TVC, the $C_{[zSC]}\delta_{TV}$ and $C_{mSC}\delta_{TV}$ terms help generate the

slightly-expanded flight envelope depicted in **Fig. 1**, and contribute to lower extractable approach speeds in landing.

Acknowledgements

We wish to thank the following individuals for encouraging and trusting us with an unorthodox reserach work: **Dr. G. Keith Richey, Mr. Douglass Bowers, Maj. T. Speers, W. Lindsay, Capt. J. Wigle, Dr. W. Calarese, Col. John Tedor**, and **Dr. Daniel W. Repperger**, all from the USAF, and last, but not least, the late **Dr. W. B. Herbst** of MBB.

References

1 - Gal-Or, B., "Vectored Propulsion, Supermaneuverability and Robot Aircraft", Springer Verlag, N.Y.-Heidelberg, 1990, 1991.

2 - Ibid., "The Fundamental Concepts of Vectored Propulsion", [AIAA] J. Propulsion, Vol. 6, No. 6, Nov-Dec., pp 747-757, 1990.

3 - Ibid., "Maximizing Post Stall, Thrust Vectoring Agility and Control Power", [AIAA] J. Aircraft. In press.

4. Ibid., "Tailless Vectored Fighters", Flight Dynamics Directorate, WPAFB, USAF/AFOSR 89-0445, July 15, 1991

5 - Baumann, D. D., "F-15B High-Angle-of-Attack Phenomena and Spin Prediction Using Bifurcation Analysis", M.Sc. Thesis, AFIT, WPAFB, Ohio, Dec. 1989, AFIT/GAE/ENY/89D-01.

6 - Herrick, P., "Fighter Aircraft Affordability, Survivability and Effectiveness through Multi-Function Thrust-Vectoring Nozzles", AIAA-89-2815. To be published in Intern. J. Turbo and Jet Engines.

7 - Goldstein, H., 'Classical Mechanics', 2nd-ed., Addison-Wesley, Reading, Mass. 1981. pp. 202-204

8 - Repperger, D. W., " Minimum Coriolis Algorithms For Motion Control", Armstrong Laboratory, WPAFB, Dayton, Ohio, 1990.

9 - Repperger, D. W., " A Study of Supermaneuverable Flight Trajectories Through Motion Field Simulation of a Centrifuge Simulator", ASME J. Dyn. Meas. and Control. In press.

Technology Limits

Technology Limit No. 1: Hot Propulsion

Unless proven vectorable inlets [with minmal distortion coefficients] for PST-TV maneuvers are available, one cannot control vectored models with jet engines in the deep PST domain without risking engine-out situations and total loss of model and its onboard computer, probes, etc.

Yet, using our new, low-distortion, vectorable inlets for this purpose, the limitation may be removed.

Suitable jet engines for this purpose are the new Teledyne 305 family of 6"-diameter engines, each costing about $ 25,000 and lasting for up to 10 hours. Their use would drive the cost of this program a few hundreds percents upwards, but they have the potential of overcoming technology limit No. 2.

Technology Limit No. 2: Cold Propulsion

Cold-jet propulsion, generated by ducted fans driven by two-stroke engines, requres no vectorable inlets and is therefore much less risky and considerably faster and more cost-effective for simulating maximum PST-TV agility and demonstrating new feasibilities of TV control power at low speeds.

However, to operate the required-size 6"-diameter ducted fans to generate sufficiently fast cold jets, one must rotate them at least as fast as 20,000 to 25,000 RPM.

Technology Limit No. 3: Piston Propulsion

Currently, there is no engine available above 5 HP which operates in the range 20,000 to 25,000 RPM. [Increasing engine HP results in reduced RPM. Hence the current technology limit is around 4 HP per engine for the 1/7-scale flying model.]

Technology Limit No. 4: Agility Measurement Affects Agility

Technology limit No. 3 limits the thrust-to-weight ratio of the flying models

While the weight of the onboard computer required to measure TV agility is only about 100 grams, the combined additional weight of gyros, extra batteries, probes, two radio controls, recording accessories, etc., is about a kilogram. This extra weight decreases the thrust-to-weight ratio beyond the minimum required for safe flight and good maneuverability. Thereby, we have encountered serious technology limitations that have caused frequent aborted takeoffs and crashes and the total loss of valuable equipment

To conclude. The very method to measure agility affects the maximum agility extractable from a TV-model based on cold propulsion

Technology Limit No. 5: Accelerometers vs. Gyros

At one point during the study we replaced the relatively heavy gyros/batteries with low-weight accelerometers. Excellent performance was obtained in the laboratory. However, when we operated the engines, the low-weight structure of the flying model introduced such vibrations that filtering them out was apparenly not effective. Hence, we had to switch back to gyros, at the cost of losing time, funds and agility.

Technology Limit No. 6: ETV Instead of the more effective ITV

Internal Thrust Vectoring [ITV] requires ducts whose area cross-section changes from circular to rectangular shape. However, such ducts, with the availble cold propulsion, causes about 33 % loss of thrust. On the other hand, our laboratory test results and the flight experience [without the gyros and instrumentation] have demonstrated that ITV provides maximum PST-TV agility for any given model.

With no solution available now to this problem, we have been forced to concentrate during the last year on External Thrust Vectoring [ETV], consisting of 4 vectoring external paddles which provide yaw and pitch thrust-vectoring control. This method does not reduce the maximum thrust available at takeoff and during climb, as do Internal Thrust-Vectoring [ITV] nozzles. However, this method provides relatively low efficiency of thrust-vectoring control power during SACOMs.

Without the additional weight of computer, gyros, batteries etc. we have thus demonstrated

the "Cobra" maneuver with ITV However, flight tests with the computer/gyros, required the use of ETV

Nevertheless, with ETV we have, so far, demonstrated at least twice the pitch rate in comparison with conventional flight control.

Technology Limit No. 7: Moments-of-Inertia, Stability Margins, Etc.

The following ratios of the moments-of-inertia of the USAF SMTD F-15 with fuel are:

Izz/Iyy = 1.15 and **Izz/Ixx = 6.25.**

In comparison, the following ratios of the moments-of-inertia of our TV F-15 model with a full fuel tank are [2% error in the measurement. Cf our Progress Report from 1990 and below]:

Izz/Iyy = 1.11 and **Izz/Ixx = 6.46.**

On one hand this good agreement provides reasonable similarity.

On the other hand, the very low moments-of-inertia values which characterize our flying models, cause amplification of air turbulence, engine vibrations, and unwanted sideslips, rolls, etc., during SACOMs.

Therefore, the results provided here for windtunnel, laboratory and flight tests should be used with caution during scaling-up procedures and scale corrections.

Our flying models are based on a +5 % static stability margin, with and without fuel. [See also the effect of fuel on the values of the moments-of-inertia of the scaled and the actual F-15s.] On the other hand, new vectored aircraft would maintain negative static stability margins and use fly-by-wire control methodologies. In addition, our flyer's hand responses [as recorded by our ground computer 43 times per second], do not scale-up. Furthermore, materials, servos, 1/7-scale TV-nozzles & engine inlets do not scale-up, or require additional empirical work prior to their adaptation to full-scale aircraft.

Technology Limit No. 8: Scaling up of Vectorable Nozzle and Inlet Test Results.

Performance test results for our yaw-pitch and roll-yaw-pitch family of TV-nozzles have been extracted from operating the nozzles installed on a 700 lbf jet engine in our "full-scale" engine test facility. These complicated "full-scale" nozzles do not scale-down to the 1/7

scale of our flying models. For this reason, and for saving weight, the yaw vanes and the pitch flaps employed for thrust-vectoring control of the flying models have been constructed from simple flat surfaces which do not correspond to the "full scale" yaw vanes and pitch flaps of the optimized TV-nozzles.

Report No. 1 to WL/FDL/WPAFB via USAF/AFOSR/EOARD[April 24, 1990] provides the calibrations of the axial, vertical and sidewise forces and moments operating on the flying models during TV-commands to deflect the jets. These data were measured under static test conditions, but when the flying-model engines operate at full throttle. We boldly assume, however, that these calibrations remain practically invariant during the dynamic flight conditions.

It should further be stressed that the geometric yaw or pitch flap deflection anglesare not the actual jet deflections. Hence, to estimate the actual forces and moments on the model during SACOMs, one must use proper calibrations of the TV-nozzles.

PST-TV Upgrading and Instrumentation Upgrading

We added, at the pilot's head location, 3 perpendicular calibrated accelerometers, and constructed and calibrated new electrical filters intended to remove engine noise from accelerometers readings.

To verify their performance under simulated laboratory conditions, we have conducted numerous laboratory simulation tests of G_x, G_y and G_z, p, q, r, α, β and V, while properly rotating and translating the PST-TV-F-15 model, with and without the engines operating.

A few such flight simulation test results are depicted in **Figs. i, ii, iii, iv, etc.**

Together with the previous instrumentation, onboard and ground computers, these accelerometers can now measure the required G_x, G_y, G_z envelopes 20 times per second.

We have also refined, especially with help provided by **USAF Capt. D. D. Baumann,** within his two **USAF/WOE** visits to this Laboratory, the methodology of this research.

Simultaneously with the conduct of laboratory and theoretical works we have developed, and verified by flight tests, an improved thrust-vectoring nozzle pitch plate [Cf. **Fig. A**].

This fast-rotating plate allows the conduction of the expanded PST-TV, rapid flip-up/down 'g-onsets' maneuvers required by this research, without compromising on instrumentation weight/capability. Unlike the previous external thrust-vectoring paddles, this one is an internal one. Hence, it provides much larger moments. The new servos and TVC mechanism installed have also considerably reduced previous delay times into effective TVC. Most important, the new plate can keep the exhaust nozzle circular, thereby not reducing the thrust by about **33%.**

During TVC the thrust is reduced momentarily by the rotating plate. However, there is no need of maximum thrust during max AoA PST-TV-'Cobra' and 'Herbst' maneuvers. [Cf. '**Tailless Vectored Fighters**', Submitted to FDD, WL, WPAFB, on July, 15, 1991.]

This TVC was developed especially for this research. For yaw TV we plan to install another TV-nozzle. The pitch-only nozzle was first flight tested on Nov. 7, 1991 in megiddo airfield.

The results have provided proof that a highly improved performance has been reached, up to about **200** deg/sec pitch rate [and up to about 110 deg pitch angle at the reversal point of the SACOM] during pseudo Cobra and Herbst maneuvers. Using our Dynamic Scale Factors [**DSF**], these rates translate to **$[200][7]^{-0.5}$ = 75 deg/sec at about 0.3-0.4M**, [and about 110 degrees pitch angle at the reversal point of the SACOM], for the full-scale F-15s. This value is about three times the maximum current pitch rate ['corner turn rate'] of the F-15As.

[**Note**: Up to about **0.3 sec delay time** has been measured in the laboratory in deflecting the jet vane of the flying model to full 20 degrees. Hence, the net pitch rate can be further increased by reducing this delay time. Higher pitch rates are also extractable by changing the TV-nozzle limits so that very rapid 30 degrees deflections become possible. Changing the static stability margin may also affect this value.]

Consequently, this new device allows us to perform new PST-TV maneuvers as well as pseudo 'Cobra' and 'Herbst' maneuvers in line with the aforementioned Tasks of this research.

Video tapes of these rapid pseudo 'Cobra' and Herbst SACOM-maneuvers are available at JPL/TIIT. Following editing and additional flight tests, we shall formally submit copies of this video tape.

Additional Upgrading

To help improve the flight testing efficiency in the field, we have purchased this month, a 386, 32Mhz, notebook computer, with 60Mb hard disk and 5 Mb internal memory. Its proper use within the framework of this research is expected to:

(i) - Prevent the dangers and time wasting associated with transportation of the model from the landing site, to Chim-Avir Hanger, about 1.5-km away, to unload the onboard computer into our 286 AT PC computer, via our Procomm computer-to-computer-communication via digital numbers.

(ii) - Allow a relatively rapid display of most critical flight test results near the runway, so as to improve instructions for next SACOM flight tests. A new programer has been hired to generate a semi-automatic post-flight display software for this purpose. Hopefully we can complete this improvement in early 1992.

To reduce the current long time delays associated with computer-to-computer-communication transfer via digital numbers in the field, we have modified the hardware of our standby onboard computer from digital to binary. This work was completed on Dec. 22, 91.

A video-converter from PAL to the American Standard NTSC has been added this month to our facilities. It is intended to improve post-flight video-taping-editing-reporting to the USAF. We have also trained the ground team and the flyer how to conduct the new expanded tasks of this work.

Following three weeks of unprecedented heavy rains, we now plan to resume flight tests with the upgraded instrumentation, following numerous flight test simulations in the laboratory and the new systems [see a few charts as examples.]

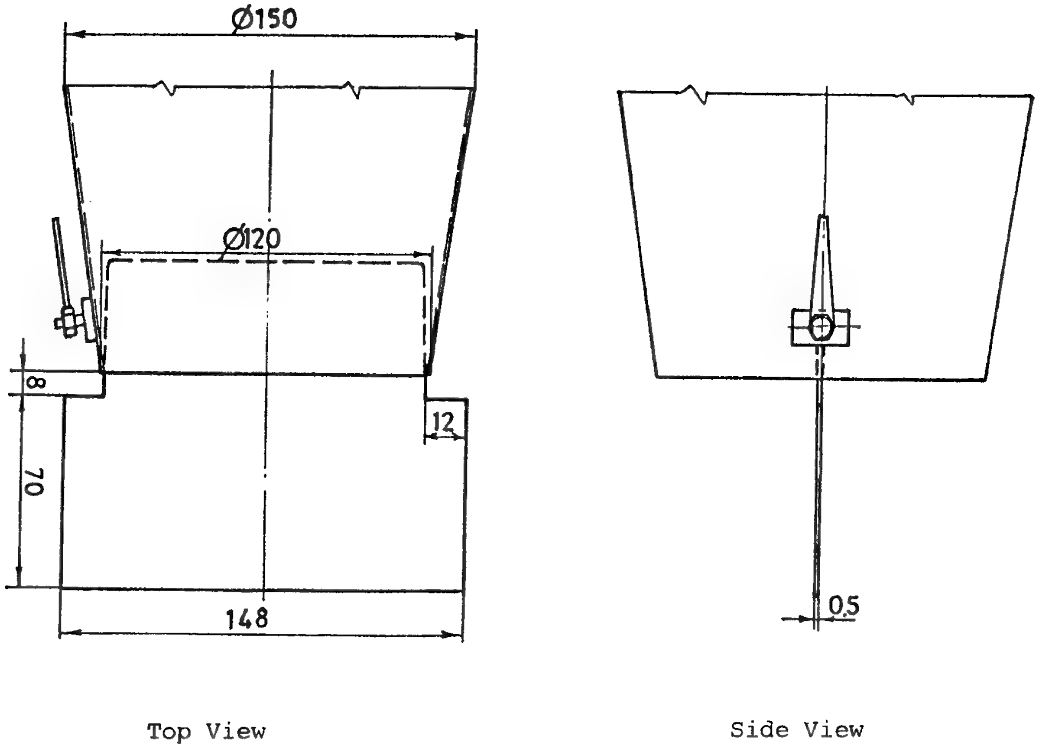


Fig. A: The pitch TV vane added to the PST-TV F-15 RPV. It has been fitted only for the Dynamic Simulation of Prototypes Using the Method of Gyration Research Project - USAF/AFOSR/EOARD - WL/AL/WPAFB, 91-92. Can be operated to 40 deg pitch angles. Currently employed with maximum deflection of 20 deg. Delay time up to 0.3 sec to 20 deg deflection. Dimensions are in mm. Depicted also is the pitch-servo arm. See text for additional details and 'Technology Limits'.
Actual operation has been recorded on video tape 7.

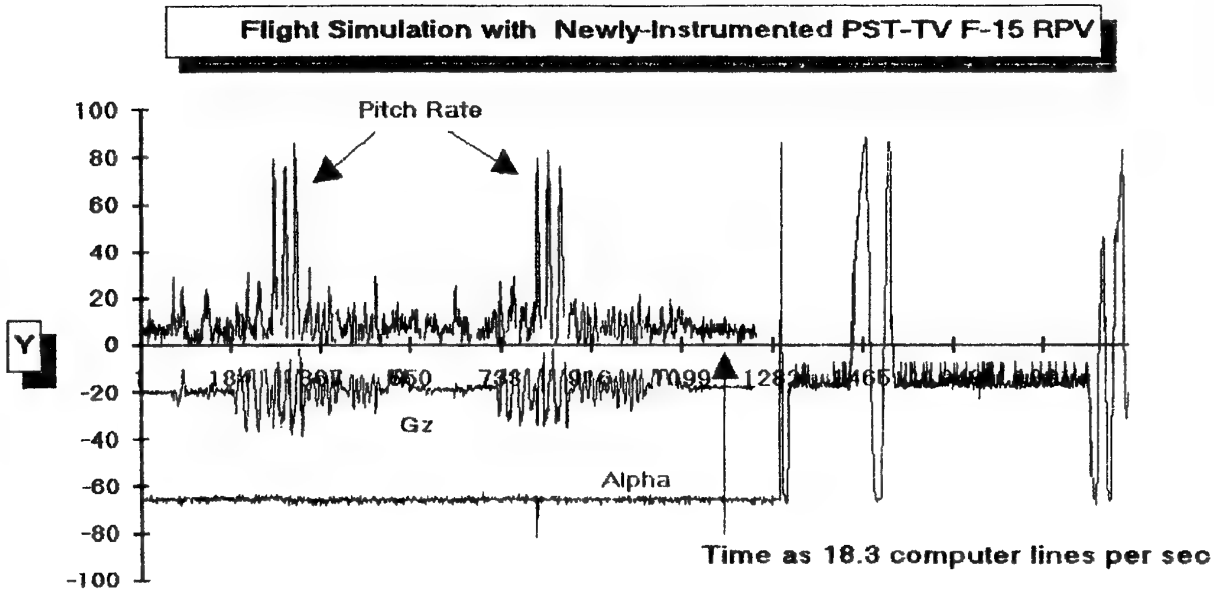


Fig. i: A Sample of the proposed new standard of reporting using our new 386 notebook computer.

Y = Pitch rate [deg/sec], or alpha [deg.], or Gz [m/Sec^2], etc., according to specific Chart.

Time is reported here as computer lines,
i.e., 18.3 lines = 1 Sec.

The simulation included :

1 - moving the PST-TV F-15 RPV up and down without pitching. This motion is shown in the Gz initial oscillations while there is no pitch rate.

2 - rotating the nose up and down. This rotation is shown in both pitch rate and Gz.

3 - The same for yaw and axial motions, and for roll.

Notes: During this simulation the alpha, beta and velocity probes were not activated. At the end of the simulation we artificially rotated the alpha probe three times, from plus 80 deg. to minus 65 deg, the maximum range expected during the next "Cobra" and Herbst maneuvers. Other variables were simultaneously recorded by the new, 3rd-generation, Flight Recording (FR) and Ground (G) computers.

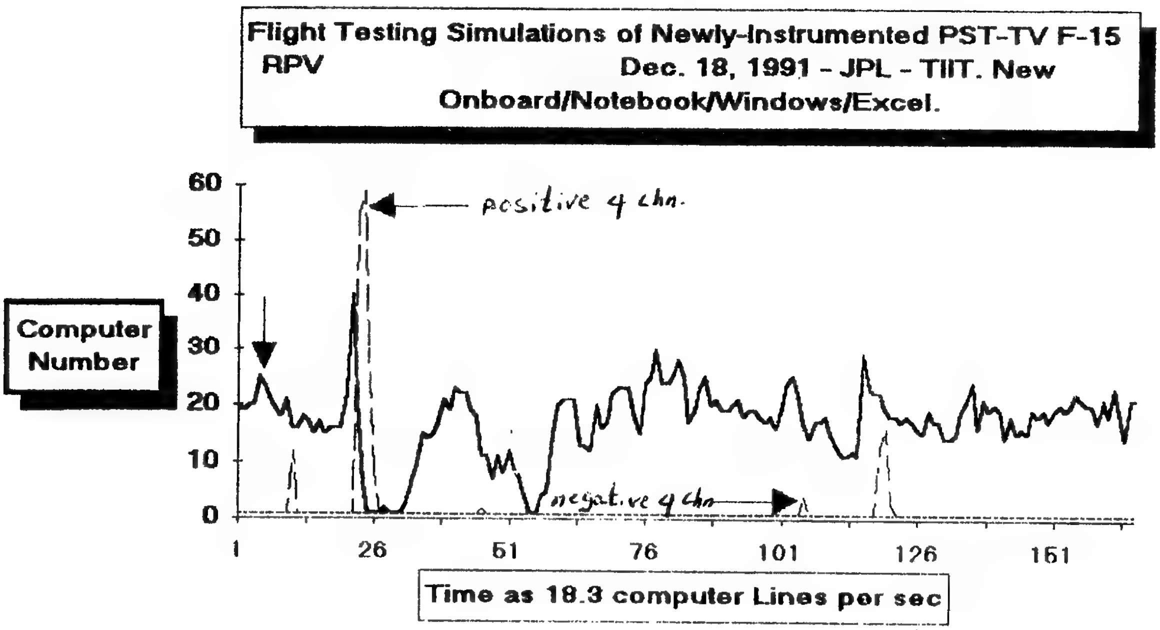
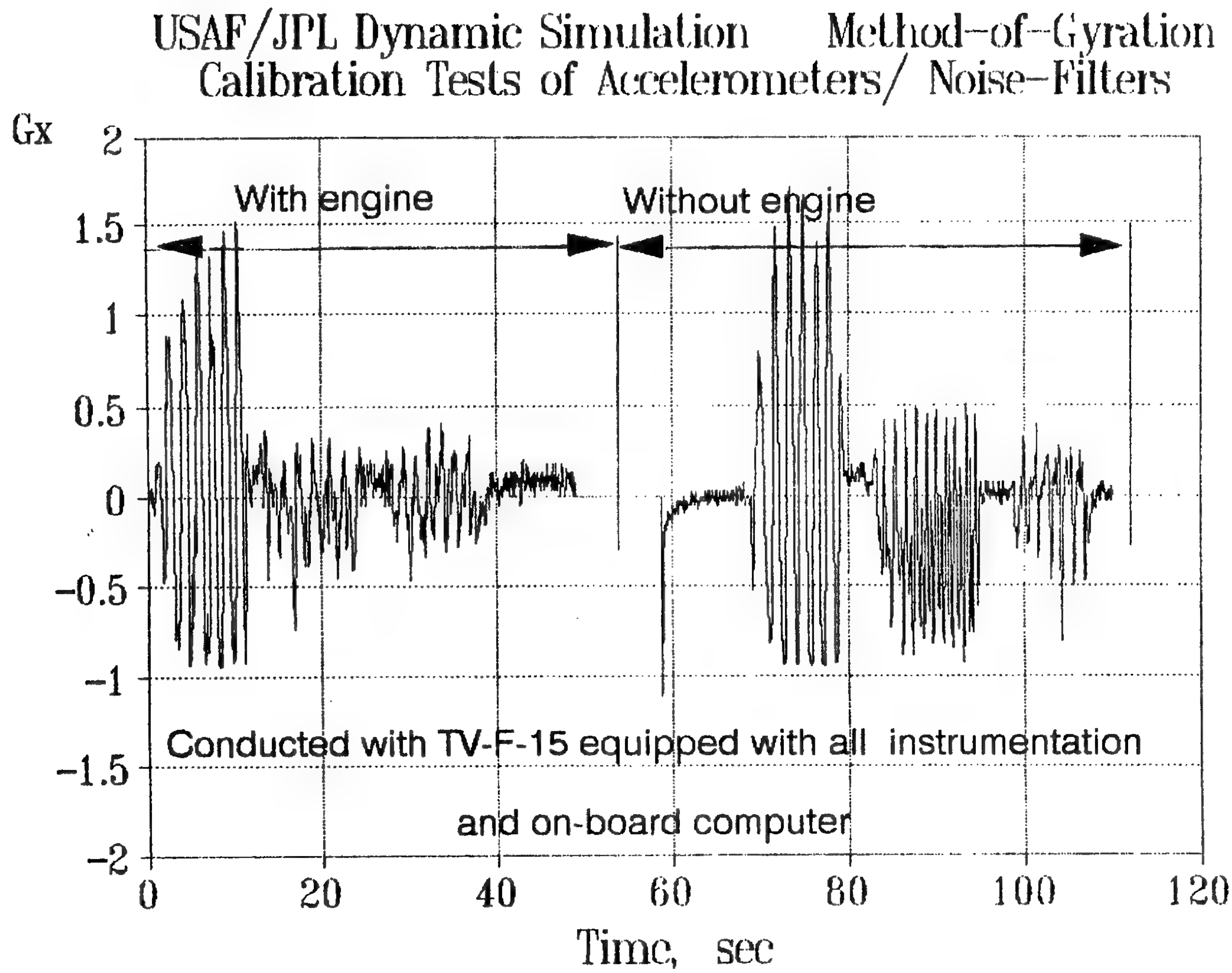


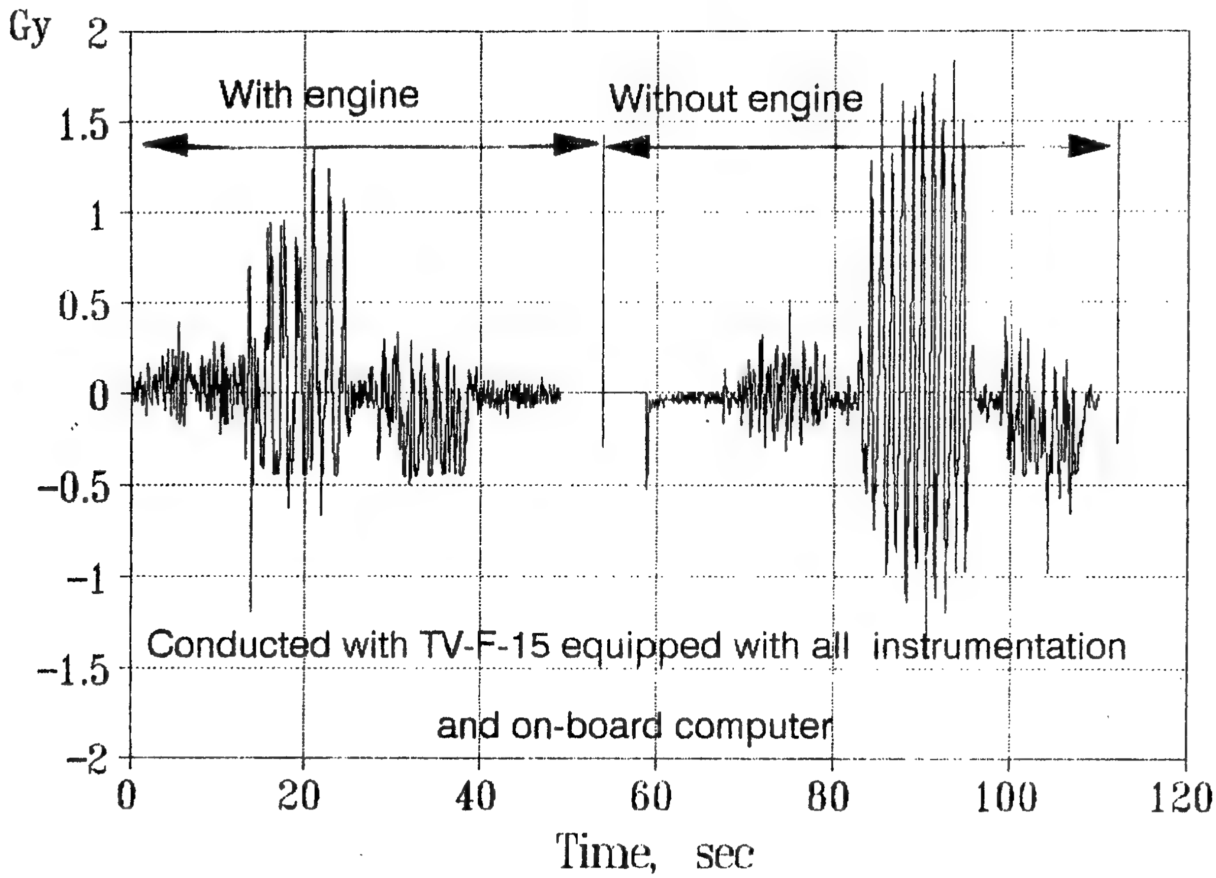
Fig. ii : A sample of "expansion chart" which can isolate a single maneuver and expand its time coordinate at will. That "window" can than be compared with the video tape. The tape can be expanded into 30 frames per sec, and, e.g., maximum pitch rate measured as deg/frames and then translated into deg/sec and compared, for verification, with the gyro-pitch rate shown on the chart of each maneuver.



Only one−engine operating. Cf. other Gi component graphs

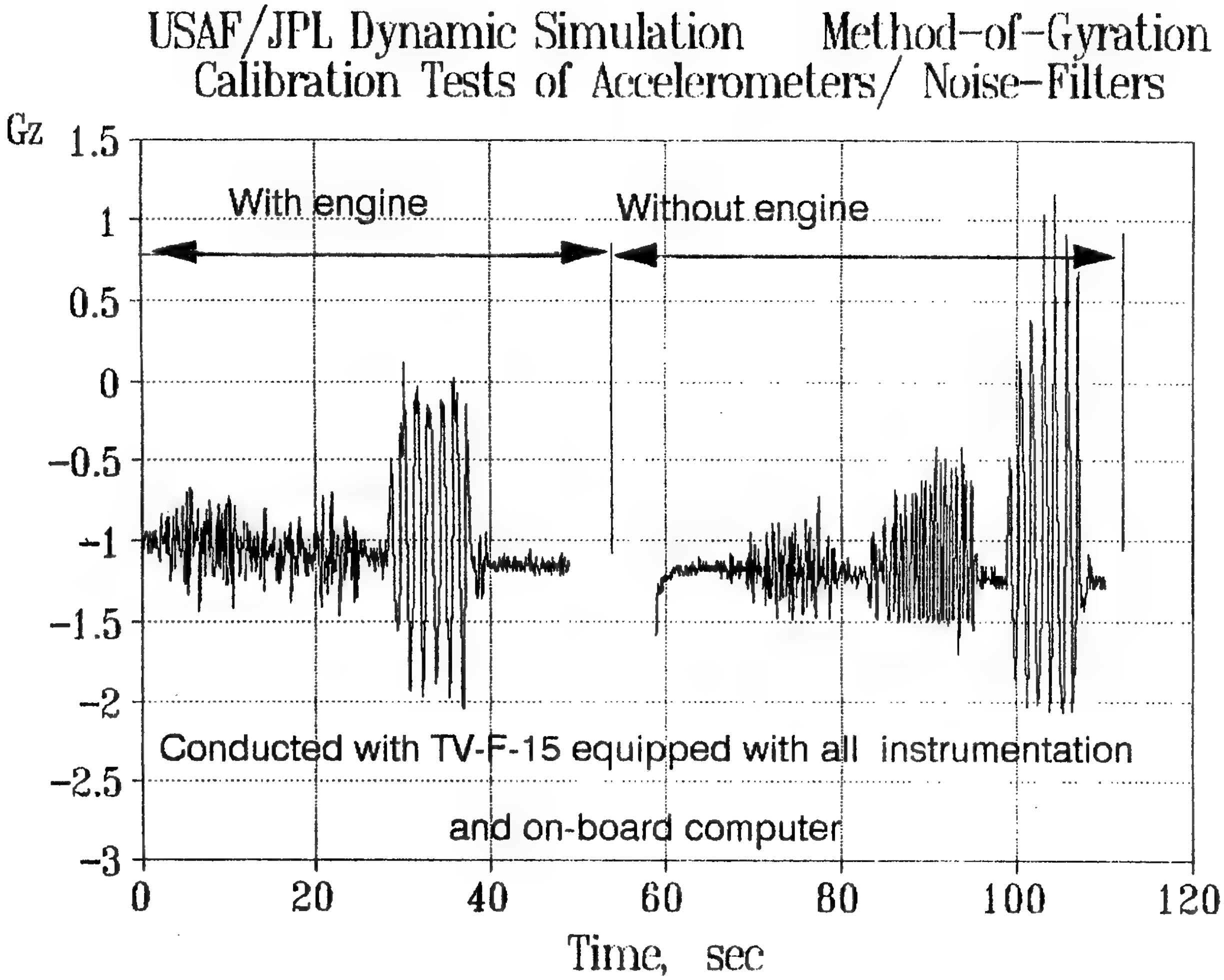
Fig. iii. cf. Appendix F.

USAF/JPL Dynamic Simulation Method-of-Gyration Calibration Tests of Accelerometers/ Noise-Filters



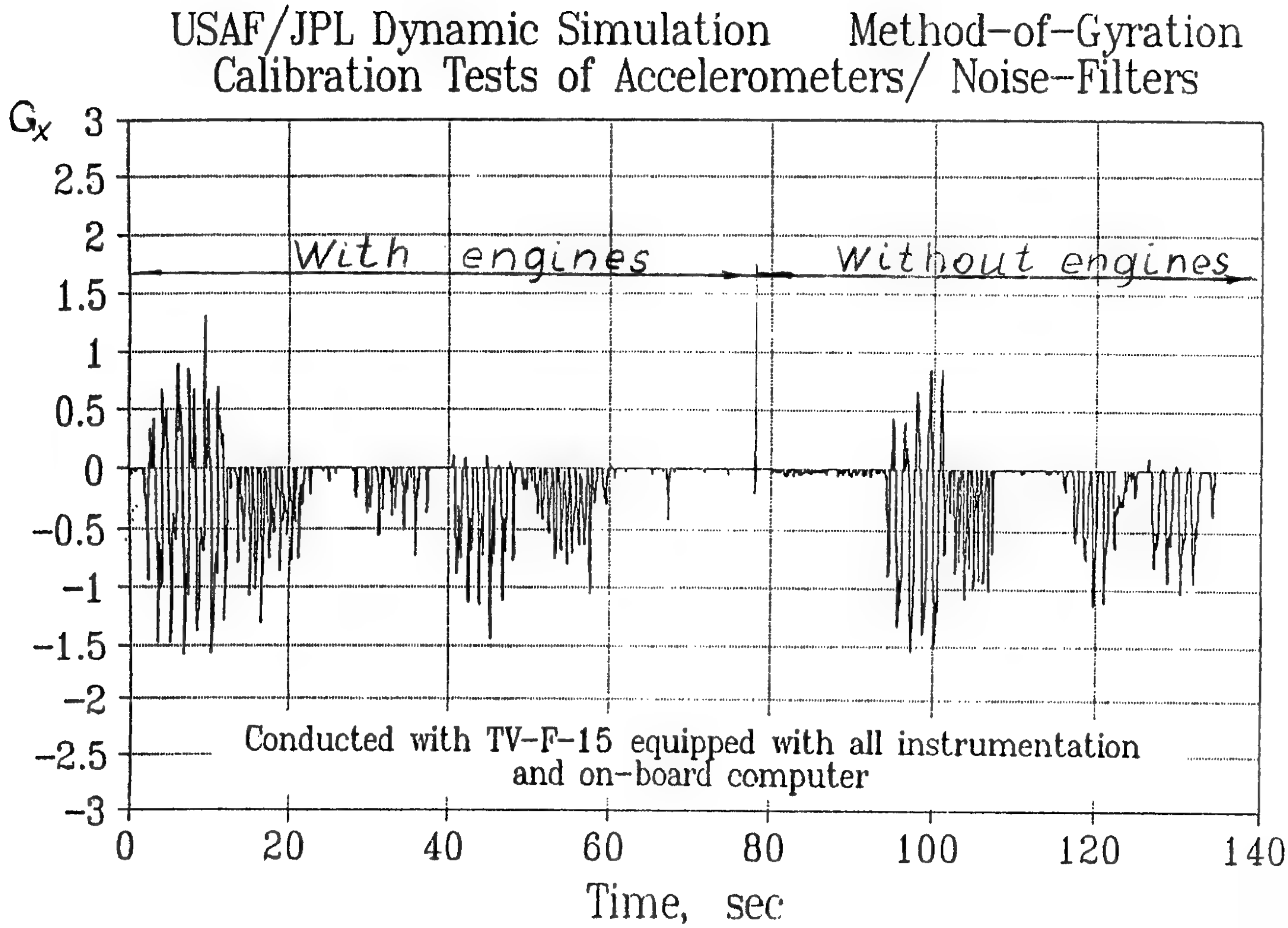
Only one-engine operating. Cf. other Gi component graphs

Fig. iv. cf. App. F.



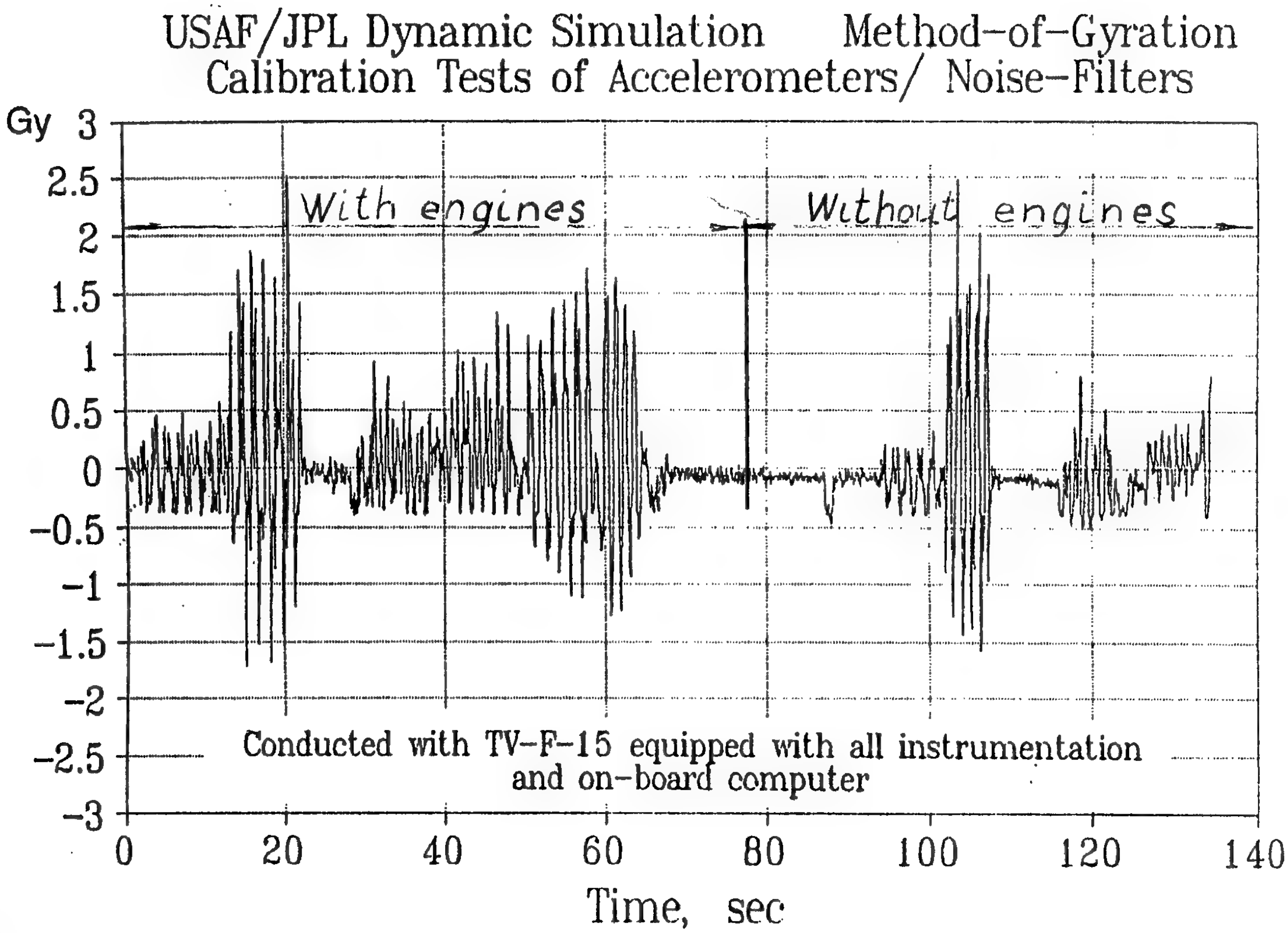
Only one-engine operating. Cf. other Gi component graphs

Fig. V. cf. App. F.



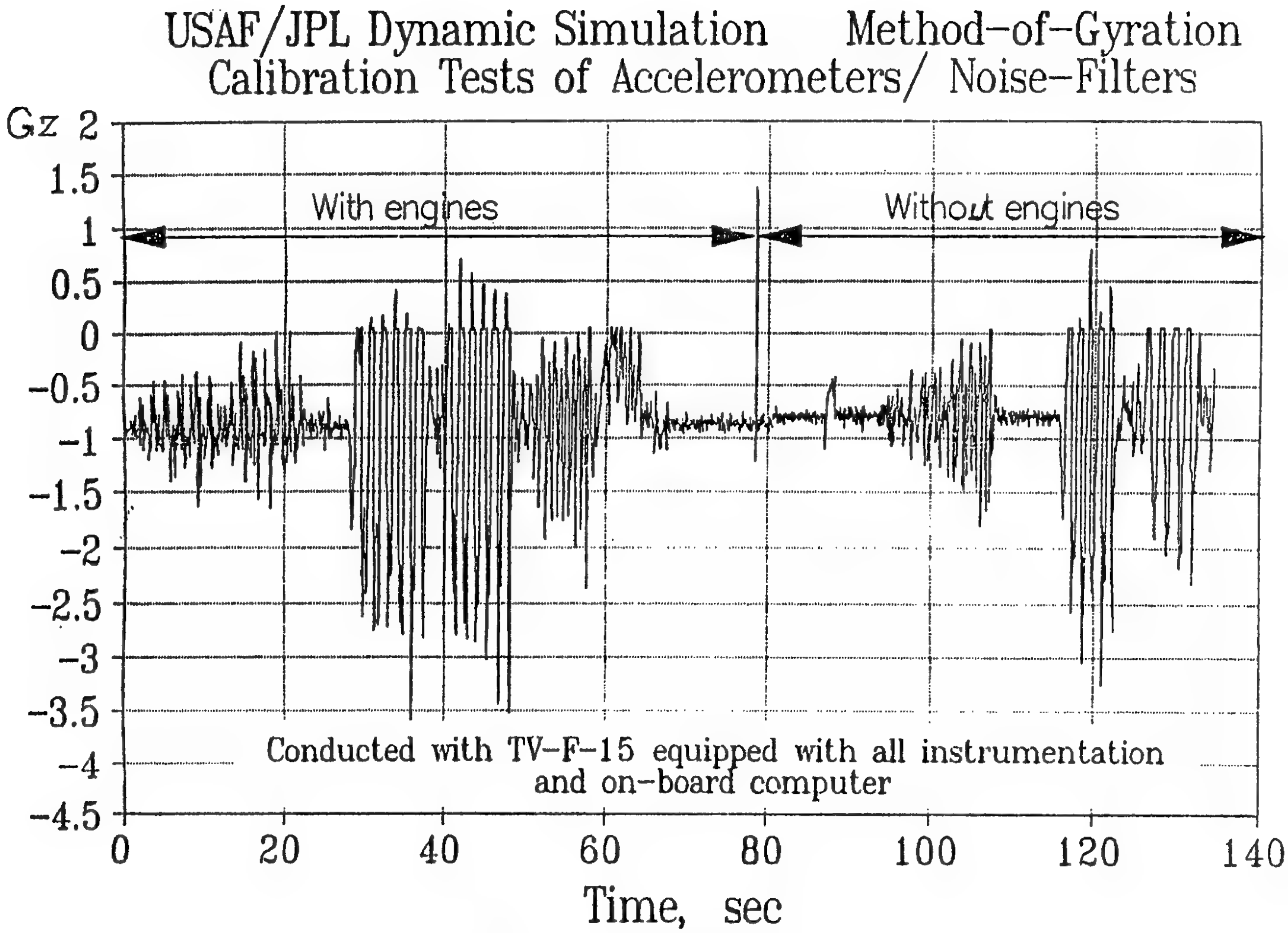
Two engines operating. Cf. other Gi component graphs

F.g. vi, cf. App. F.



Two engines operating. Cf. other Gi component graphs

Fig. vii. cf. App. F.



Two engines operating. Cf. other Gi component graphs

Fig. viii, cf. App. F

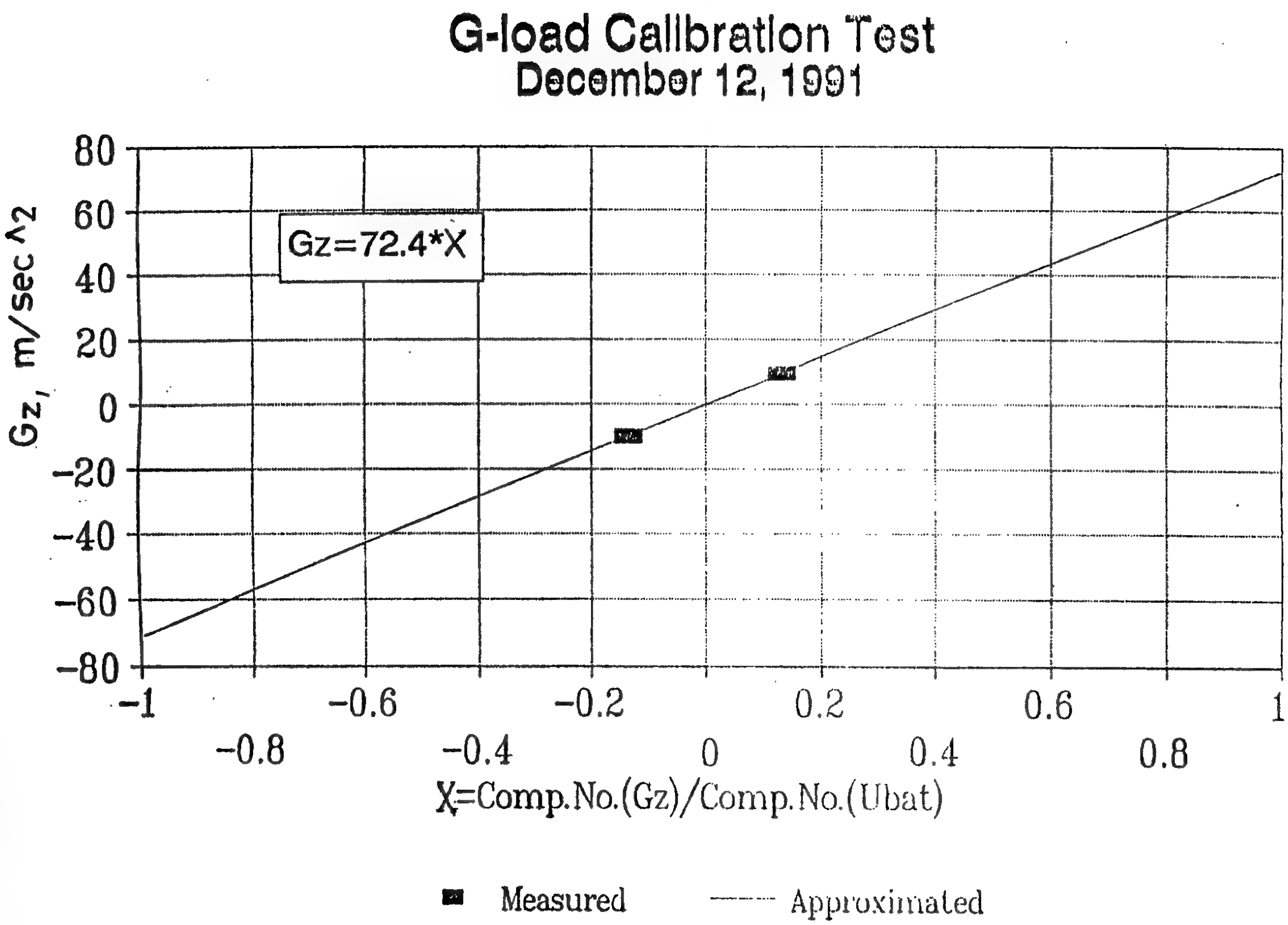
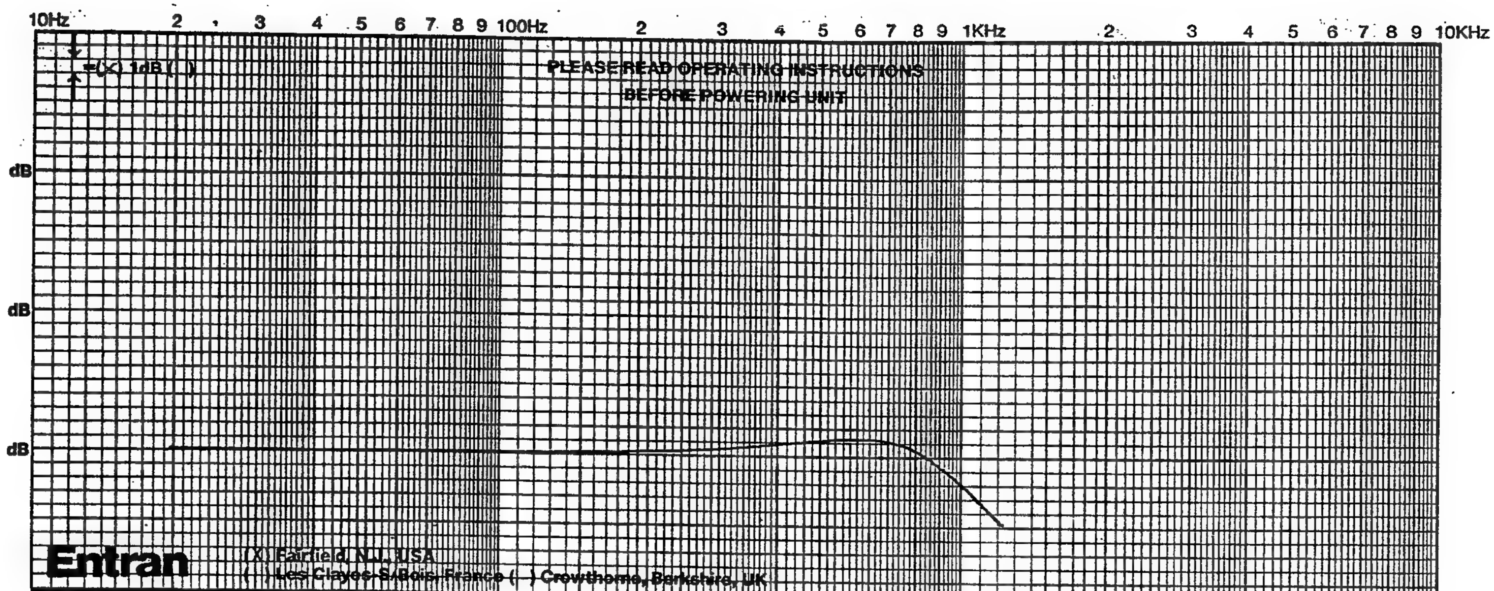


Fig. ix. cf. p. 33a

33a



Entran (X) Fairfield, N.J. USA
( ) Les Clayes-S/Bois, France ( ) Crowthorne, Berkshire, UK

SPECIFICATIONS

Model : EGA-125-10D
S/N: 90B89K04-G19
Range : ±10g Limit: ±50g
Axis:
Temperature Range, Compensated: 70 to 170°F Operating: -40 to 250°F

G_z - CALIBRATION DATA

Subject to 10,000g Shock in: ( ) Sensitive Axis Only ( ) All Axes

1,2 Non-Linearity :	±1%	1 Hysteresis:		2 X-axis :	
1 Zero :		1 Therm. Z :	±1%FS/100°F	2 TSS :	±2.5%/100°F
R Cal :		with :		across :	
Output :	9.07 mv/g	with In :	15.0 VDC	Max. In :	18.0 VDC
Ref. Temp :	75°F	Ref. Hz :	50	4 Approx. In.:	
Tested with:				S/N:	
Output :		with In :		Max. In :	
3 Input Ω :	801	3 Output Ω :	512	See Over : ( )	

ACCELEROMETER WIRING

To Module: Total:
+In: (X) Red ( ) +Out: (X) Grn ( )
( ) Grn & Blue
-In: (X) Blk ( ) -Out: (X) Wht ( )
Connector: Mate:

ELECTRONICS

Signal In Conn :	Mate :
+Sensor Power :	+Signal In :
-Sensor Power :	-Signal In :
Amp Out Conn :	Mate :
+Supply :	+Signal Out :
-Supply :	-Signal Out :
(For ±15V Input, Do not use unipolar 30V power supply)	Common :

G-load Calibration Test
December 12, 1991

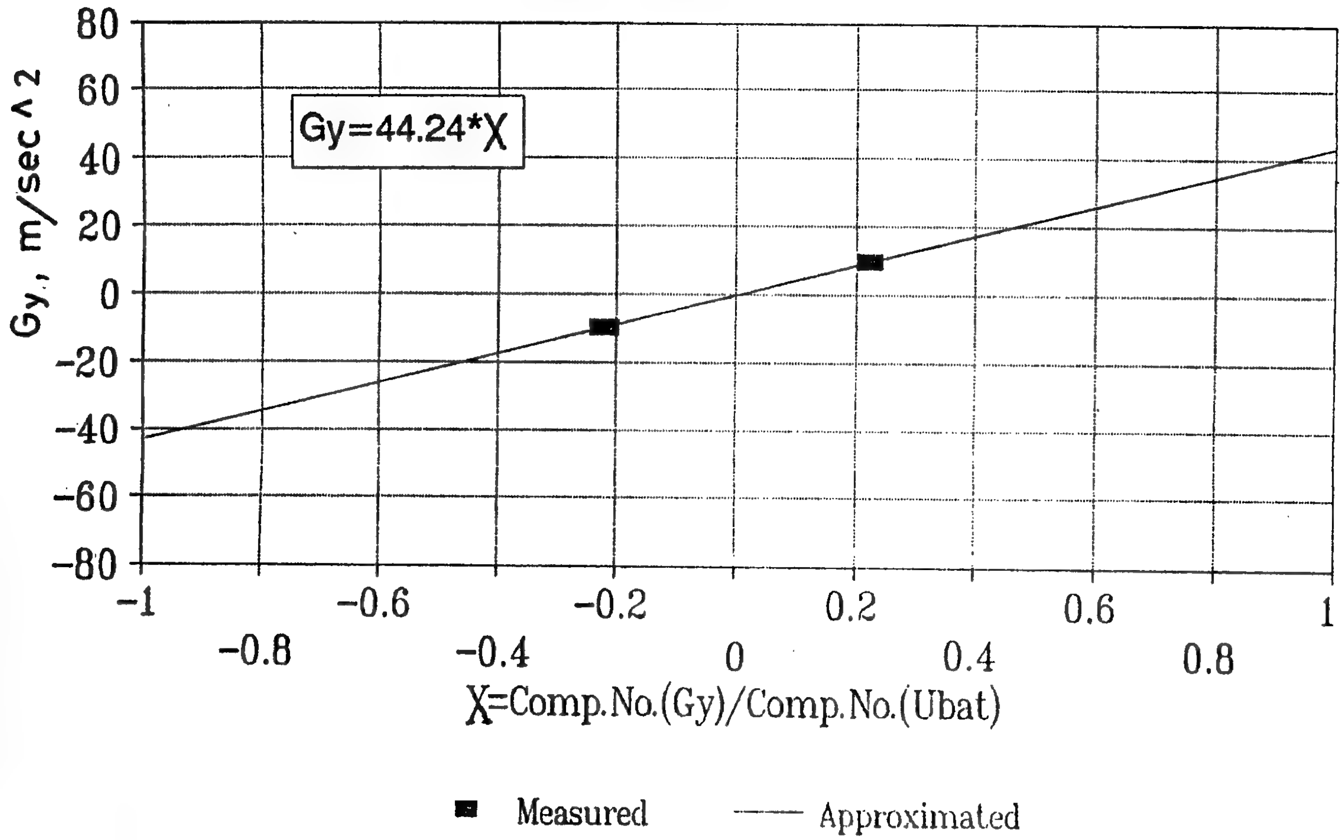
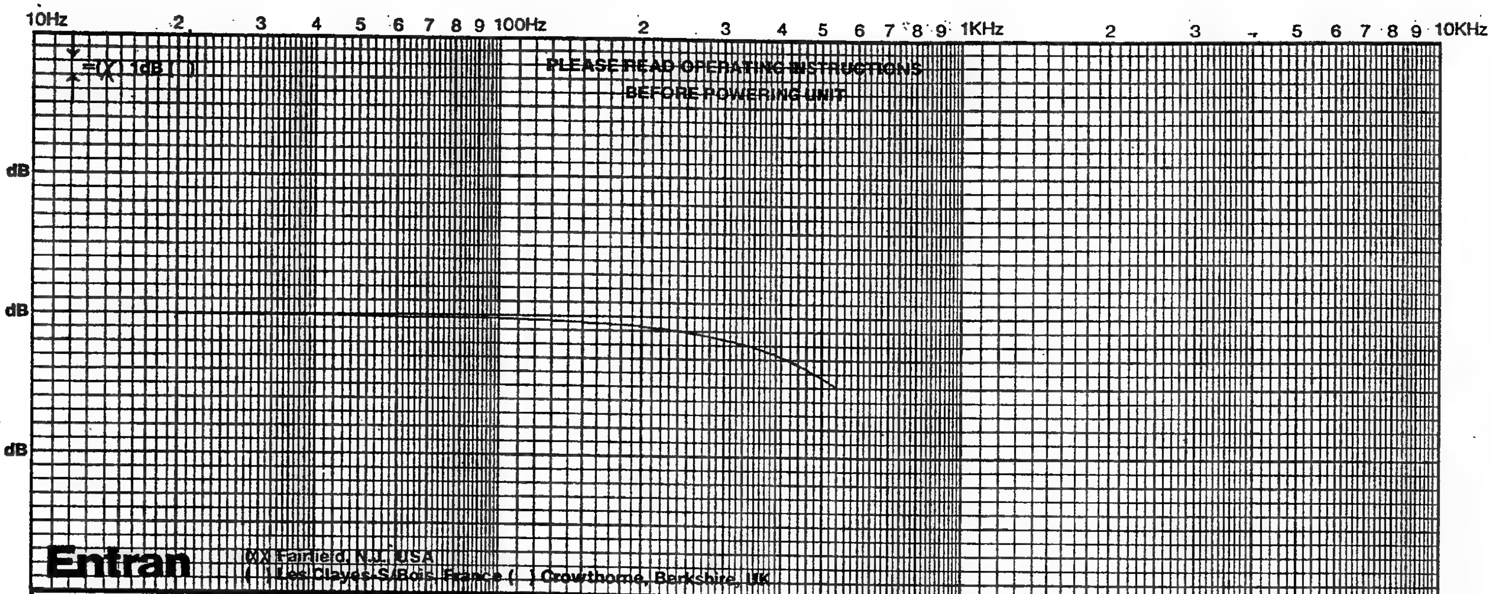


Fig. x. cf. p. 34a.



Entran (X) Fairfield, N.J. USA
( ) Les Clayes-S/Bois, France ( ) Crowthorne, Berkshire, UK

SPECIFICATIONS

Model : EGA-125-10D
Range : ±10g
Limit: ±50g
Temperature Range, Compensated: 70 to170°F
S/N: 90B 89K24-G14
Axis:
Operating: -40 to 250°F

G_y – CALIBRATION DATA

Subject to 10,000g Shock in: ( ) Sensitive Axis Only ( ) All Axes

1,2 Non-Linearity: ±1%
1 Zero :
R Cal :
Output : 12.30mv/g
Ref. Temp : 75F
Tested with:
Output :
3 Input Ω : 858

1 Hysteresis:
1 Therm. Z : ±1%FS/100°F
with :
with In : 15.00VDC
Ref. Hz : 50
with In :
3 Output Ω : 463

2 X-axis :
2 TSS : ±2.5%/100°F
across :
Max. In : 18.0 VDC
4 Approx. fn.:
S/N:
Max. In :
See Over :( )

ACCELEROMETER WIRING

To Module:
+In: (X) Red ( )
-In: (X) Blk ( )
Connector:

Total:
+Out: (X) Grn ( )
( ) Grn & Blue
-Out: (X) Wht ( )
Mate:

ELECTRONICS

Signal In Com :
+Sensor Power :
-Sensor Power :
Amp Out Com :
+Supply :
-Supply :
(For ±15V Input, Do not use unipolar 30V power supply)

Mate :
+Signal In :
-Signal In :
Mate :
+Signal Out :
-Signal Out :
Common :

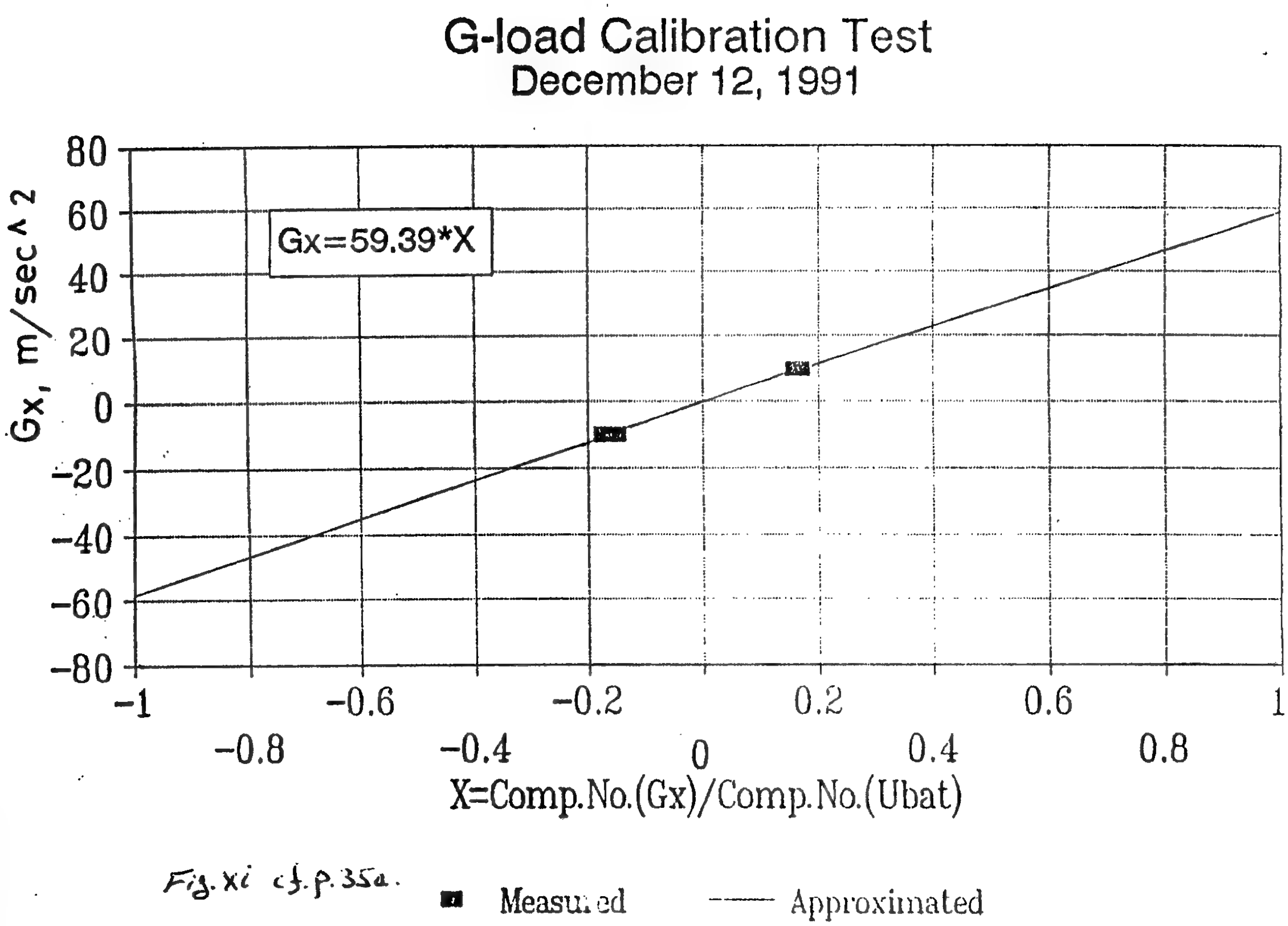
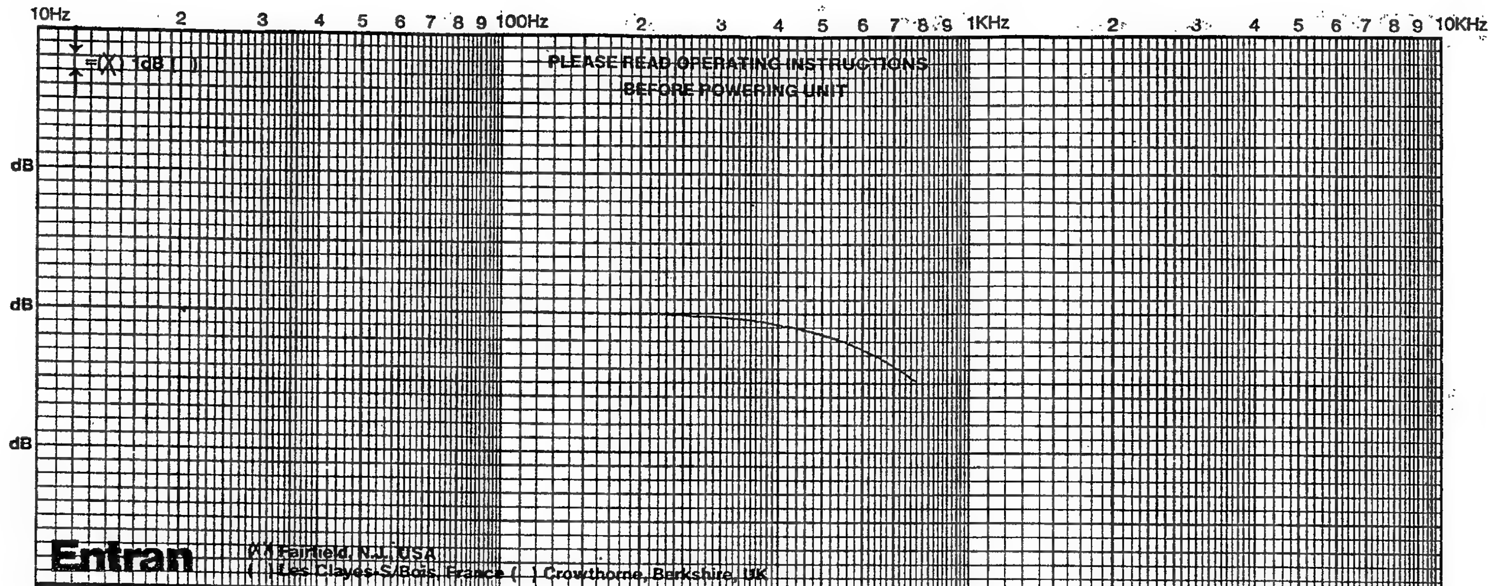


Fig. xi cf. p. 35a.



SPECIFICATIONS

Model : EGA-125-10D

Range : ±10g

Limit: ±50g

Temperature Range, Compensated: 70 to 170°F

S/N: 90B89K24-G20

Axis:

Operating: -40 to 250°F

G_x - CALIBRATION DATA

Subject to 10,000g Shock in: ( ) Sensitive Axis Only ( ) All Axes

1,2 Non-Linearity : ±1%

1 Zero :

R Cal :

Output : 9.75mv/g

Ref. Temp : 75F

Tested with:

Output :

3 Input Ω : 811

1 Hysteresis:

1 Therm. Z : ±1%FS/100°F

with :

with in : 15.00VDC

Ref. Hz : 50

with in :

3 Output Ω : 480

2 X-axis :

2 TSS : ±2.5%/100°F

across :

Max. in : 18.0 VDC

4 Approx. in.:

S/N:

Max. in :

See Over : ( )

ACCELEROMETER WIRING

To Module:

+In: (X) Red ( )

-In: (X) Blk ( )

Connector:

Total:

+Out: (X) Grn ( )

( ) Grn & Blue

-Out: (X) Wht ( )

Mate:

ELECTRONICS

Signal In Com :

+Sensor Power :

-Sensor Power :

Amp Out Com :

+Supply :

-Supply :

(For ±15V Input, Do not use unipolar 30V power supply)

Mate :

+Signal In :

-Signal In :

Mate :

+Signal Out :

-Signal Out :

Common :

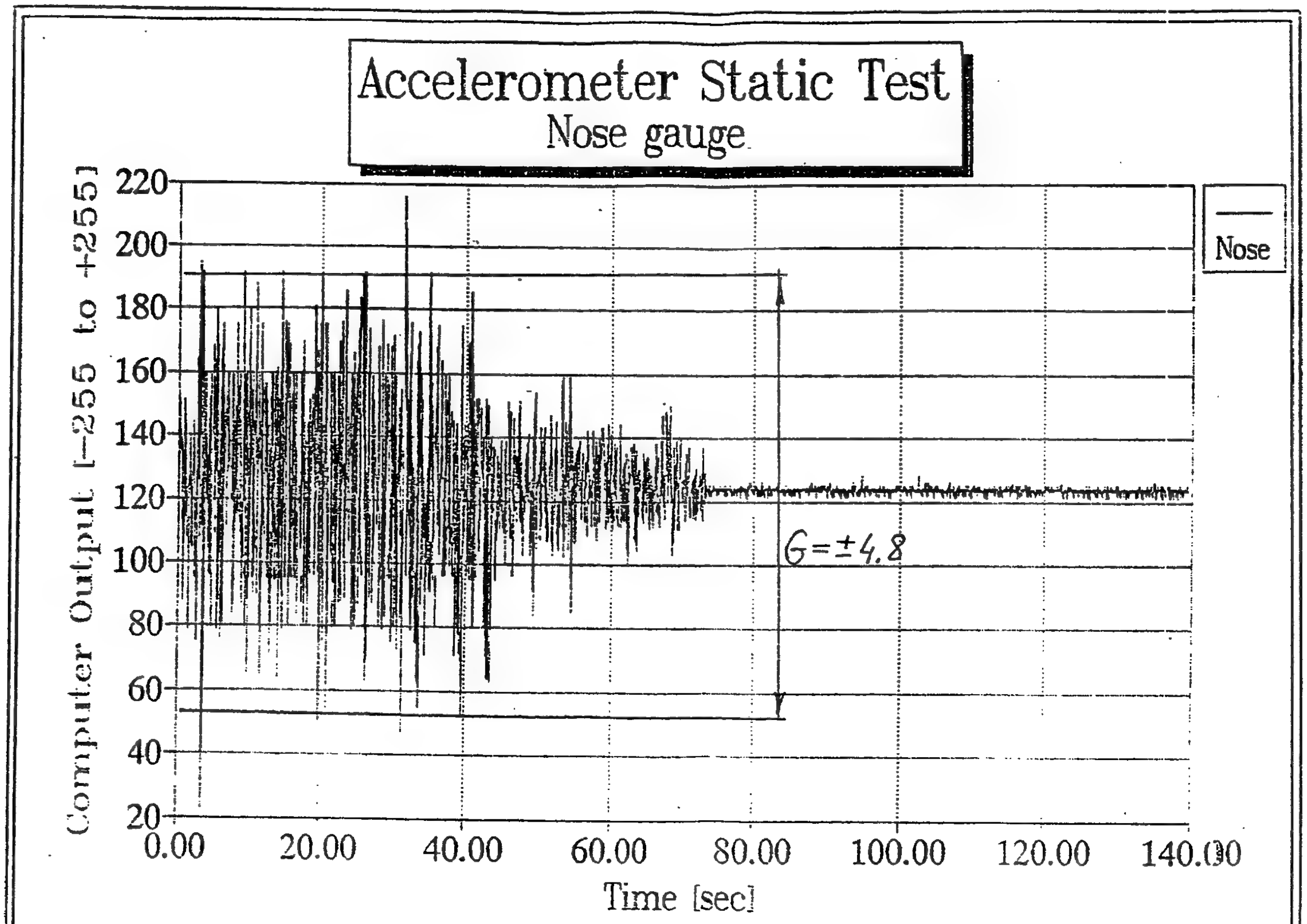
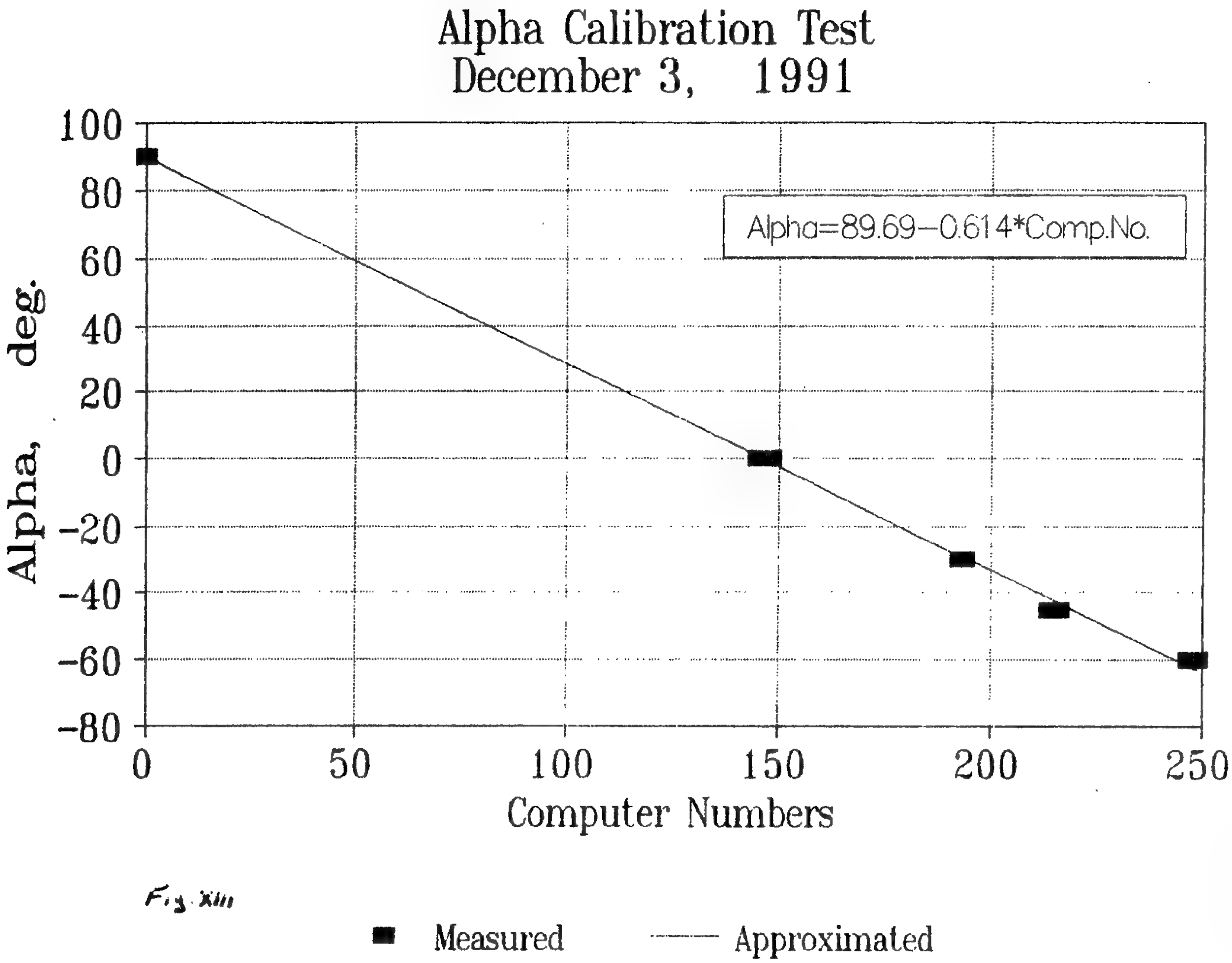


Fig. xii Cf. App. F



Alpha Calibration Test
December 3, 1991
Alpha=89.69-0.614*Comp.No.
Alpha, deg.
Computer Numbers
100
80
60
40
20
0
-20
-40
-60
-80
0
50
100
150
200
250
Measured
Approximated

Fig. XIII

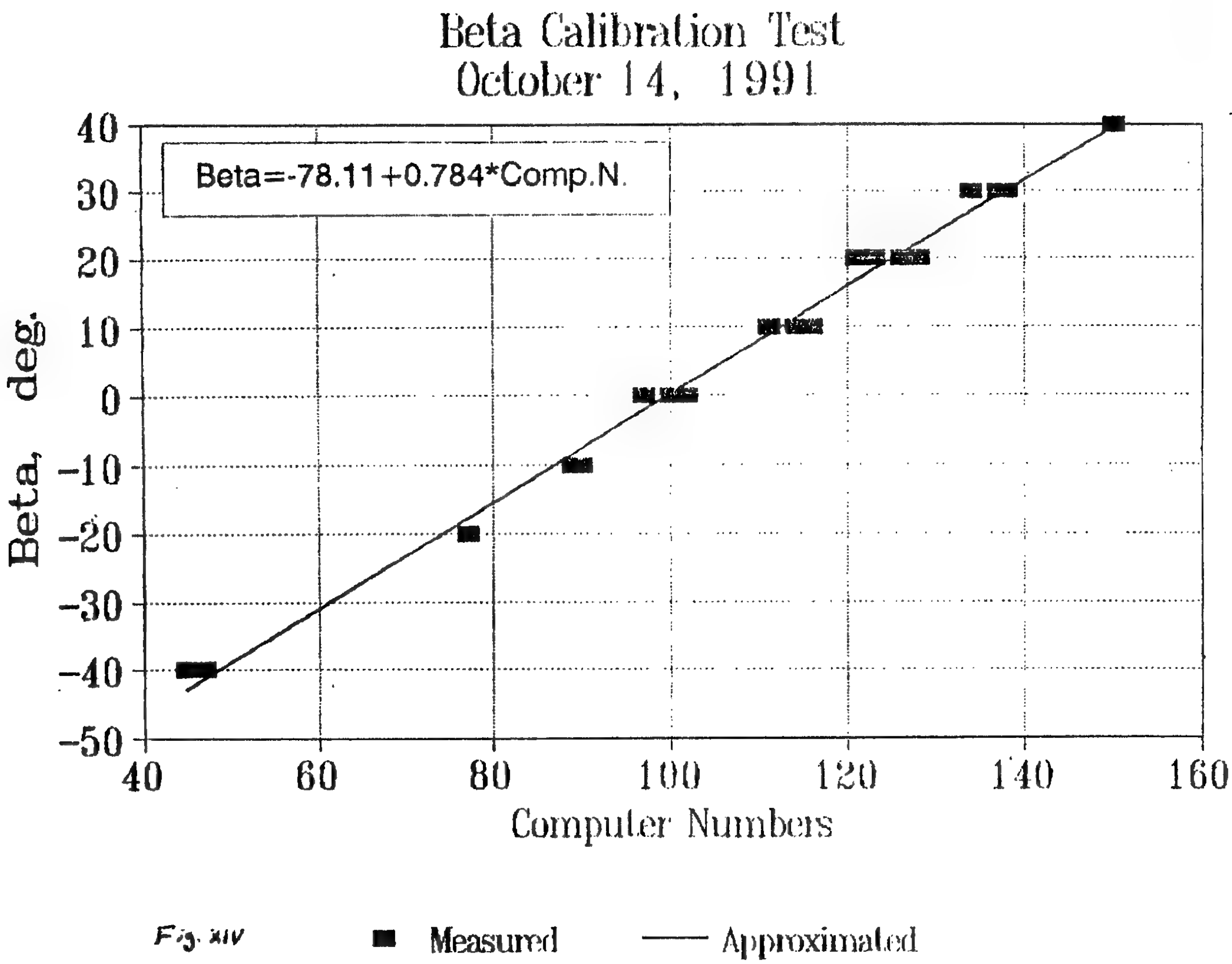

Beta Calibration Test
October 14, 1991
Beta=-78.11+0.784*Comp.N.
Beta, deg.
40
30
20
10
0
-10
-20
-30
-40
-50
40
60
80
100
120
140
160
Computer Numbers
Measured
Approximated

Fig. XIV

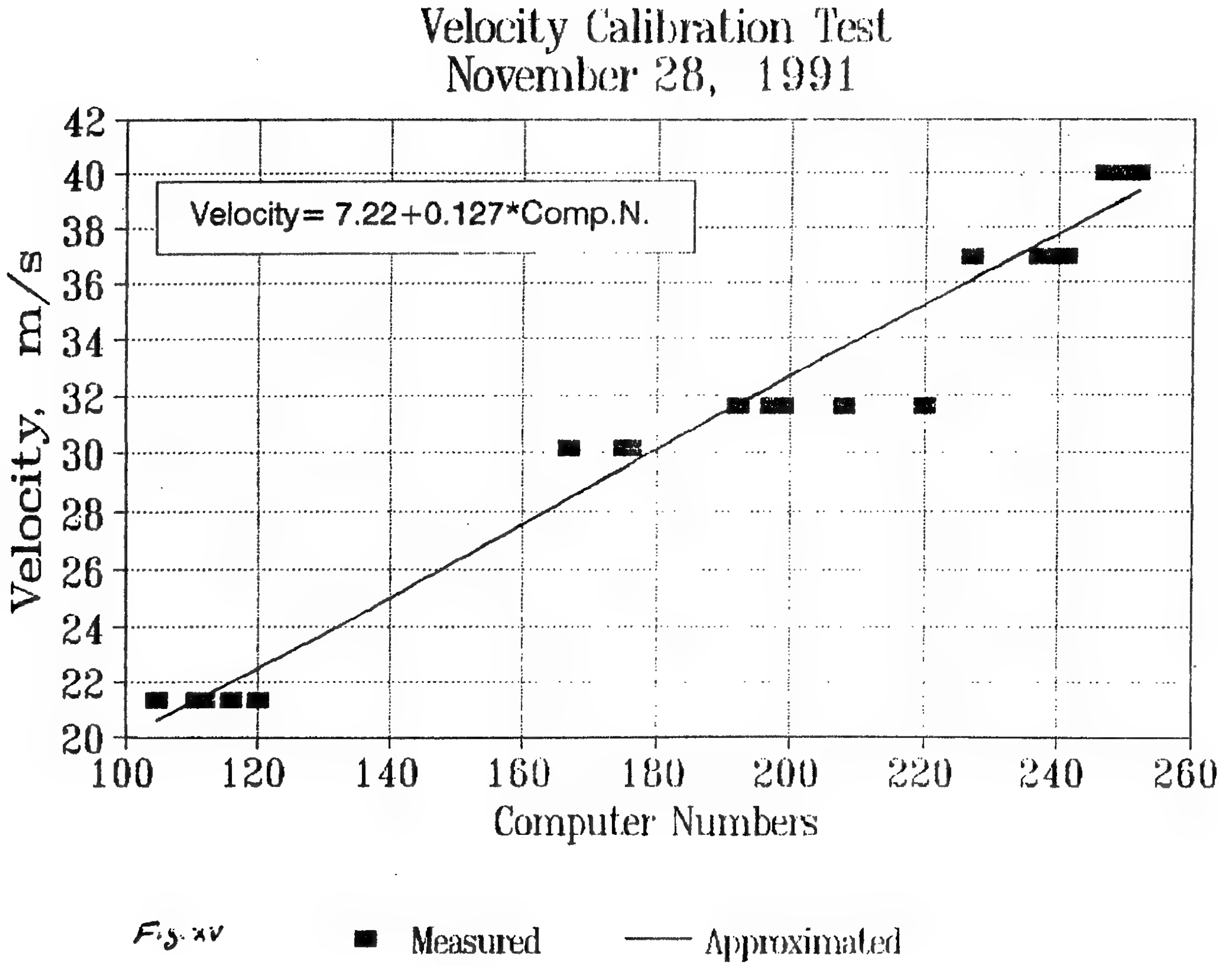

Velocity Calibration Test
November 28, 1991
Velocity= 7.22+0.127*Comp.N.
Velocity, m/s
42
40
38
36
34
32
30
28
26
24
22
20
100
120
140
160
180
200
220
240
260
Computer Numbers
Measured
Approximated

Fig. XV

APPENDIX A

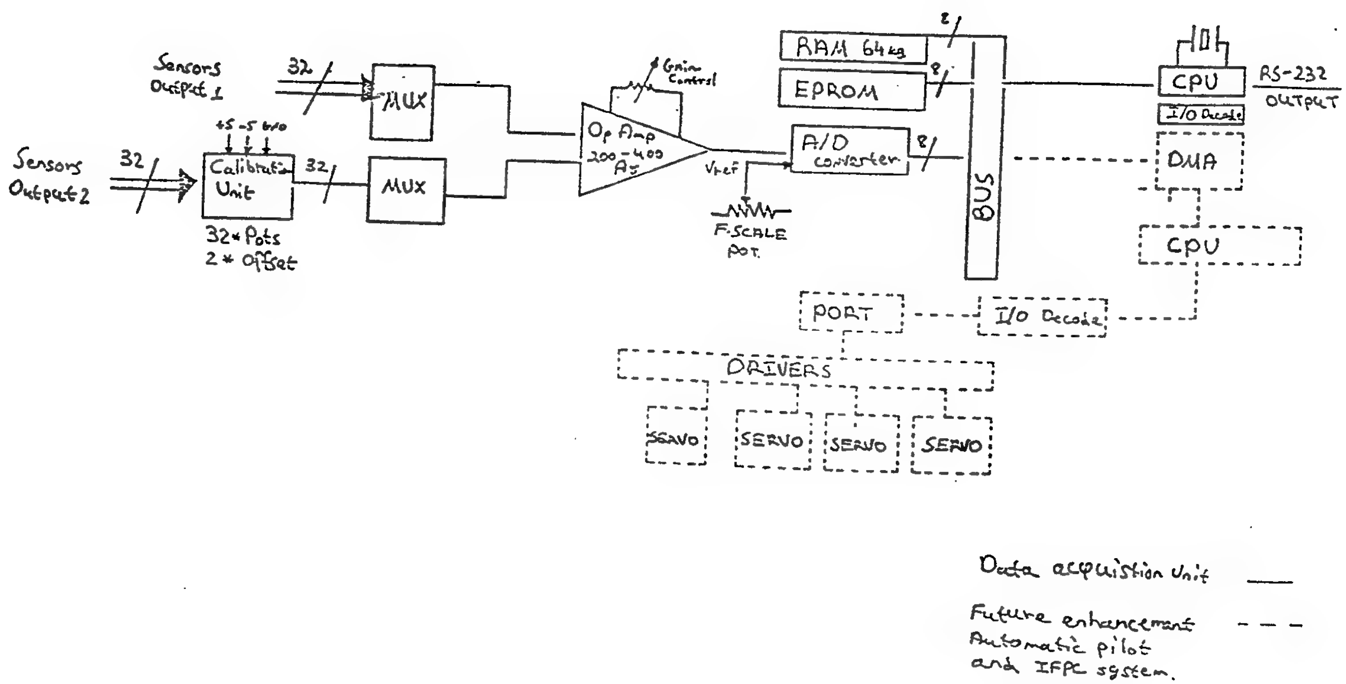
Full Documentation of the Design Criteria

of our

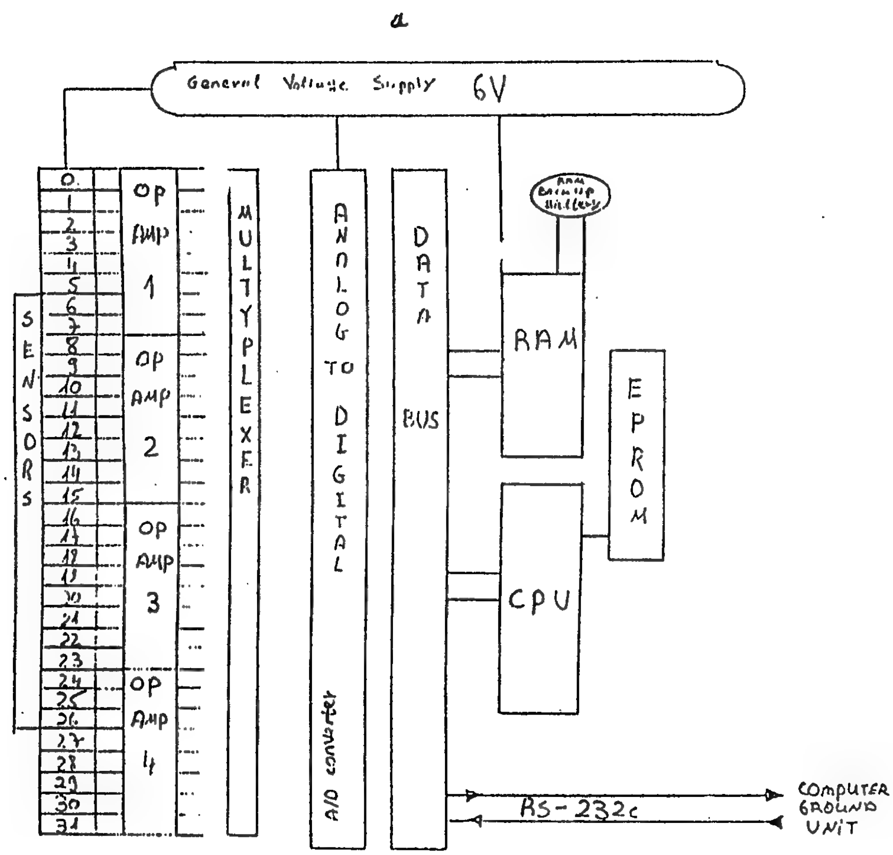
Flight Recording and Ground Computers

and their

Associated Software



On-Board Data-Acquisition Computer Function Diagram.



On-Board Data-Acquisition Computer Block-Diagram.

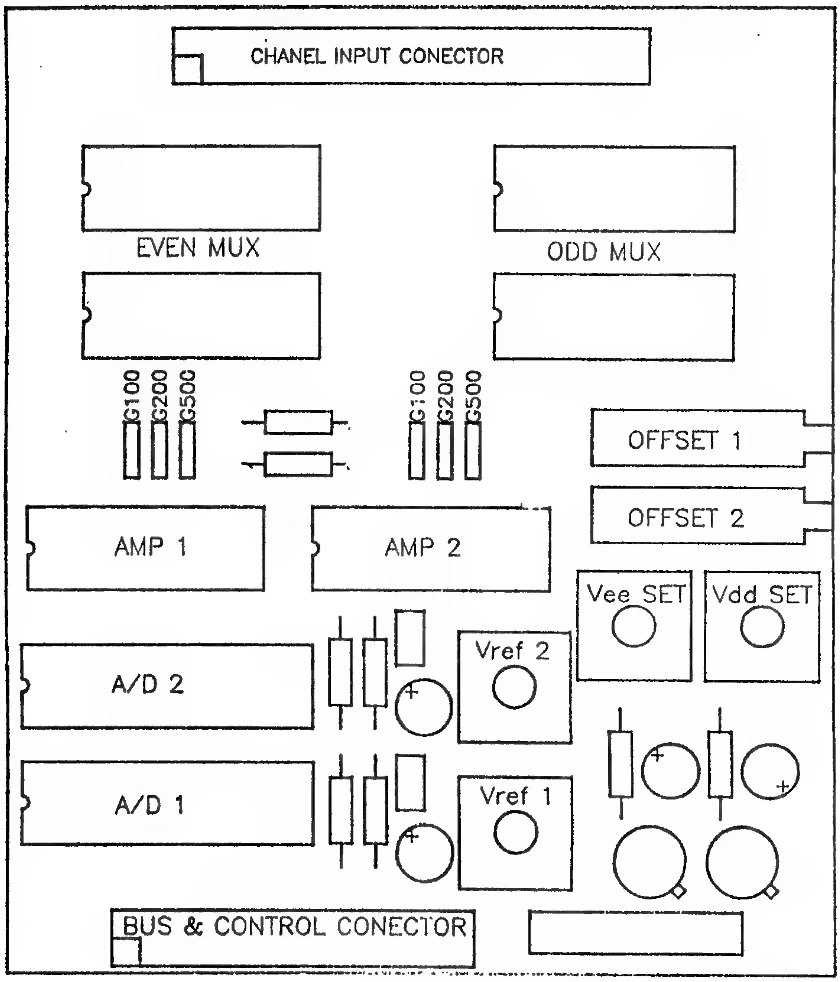
The present Vectored RPV program calls for three computerized data-acquisition systems :

(1) On-board data-acquisition computer using sensors No. 0-10 (including one switching channel for remote control system activation).

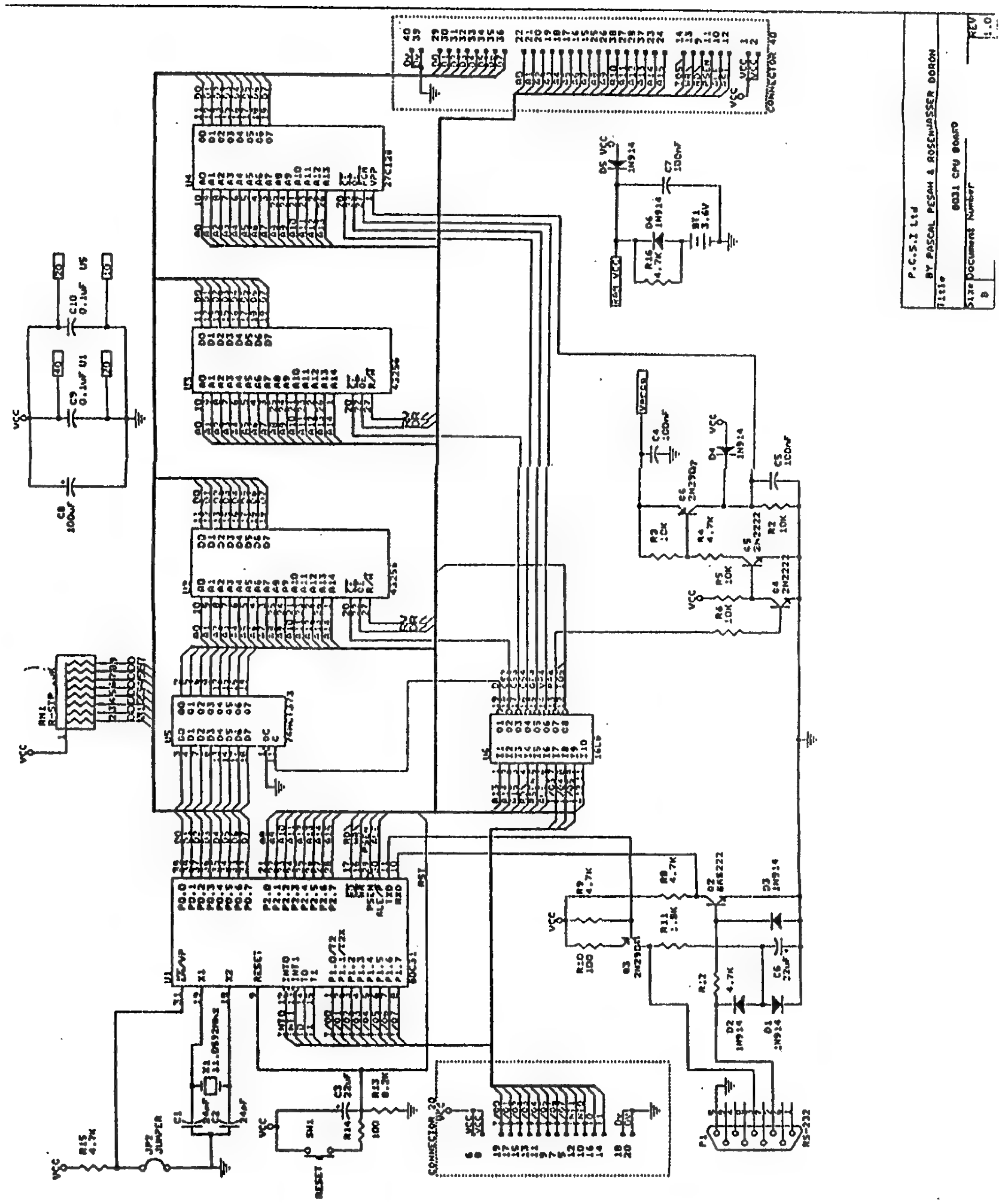
(2) Runway-based transmitter-linked computer for RPV control-input recording, using 10 input channels.

(3) IBM PC AT plus a special software package which has been developed by this lab, for transferring and analysis of the flight data, recorded by the on-board computer.

b


CHANEL INPUT CONECTOR
EVEN MUX
ODD MUX
G100
G200
G500
G100
G200
G500
OFFSET 1
OFFSET 2
AMP 1
AMP 2
Vee SET
Vdd SET
Vref 2
A/D 2
A/D 1
Vref 1
BUS & CONTROL CONECTOR

c


P.C.S.I Ltd
BY PASCAL PESAH & ROSENWASSER DORON
Title
8031 CPU BOARD
Size
Document Number
B
REV
1.0

```
;-----------------------------------------------------------------------;
;                                                                       ;
;       8031 CPU SINGLE BOARD COMPUTER MONITOR ROM                      ;
;       V-3                                                             ;
;       BY PASCAL PESAH & ROSENWASSER DORON                             ;
;       P.C.S.I service & development laboretries Ltd. ISRAEL          ;
;                                                                       ;
;       THIS PROGRAM IS A SMALL MONITOR FOR THE INTEL 8031              ;
;       IT PROVIDES A MINIMUM LEVEL OF UTILITY FUNCTION FOR             ;
;       THE USER EMPLOYING INTER-ACTIVE CONSOLE VIA RS-232              ;
;                                                                       ;
;-----------------------------------------------------------------------;
                BSEG AT(00H)
XON:            DBIT    1
IN:             DBIT    1
CH:             DBIT    1


                DSEG AT(30H)
DLY:            DS      1
TMP:            DS      1
D1:             DS      1
D2:             DS      1


                CSEG AT(00H)
                JMP     BEGIN


                CSEG AT(23H)            ;SERIAL INTERUPT
SER_INT:        JMP     INT_IN

CR              EQU     0DH
LF              EQU     0AH
TAB             EQU     09H             ;TAB
STK             EQU     48H             ;STACK ADDRESS

                CSEG AT(30H)
BEGIN:          SJMP    START

START1:         NOP
                NOP
                NOP
                NOP
                NOP
                SJMP    START


START:          MOV     IE,#00H
                MOV     SP,#STK         ;STACK = 40 HEX
                MOV     TMOD,#00H
                MOV     TCON,#00H
                MOV     IP,#00H

        ;INITIALIZE SERIAL PORT FOR 4800 BAUD RATE

                MOV     TMOD,#00100000B ;TIMER 1 8 BIT AUTO RELOAD
                                        ;150  BAUD -> TH1=#040H
                                        ;300  BAUD -> TH1=#0A0H
                                        ;600  BAUD -> TH1=#0D0H
```

```
                                        ;1200 BAUD -> TH1=#0E8H
                                        ;2400 BAUD -> TH1=#0E8H
             MOV      TH1,#0FAH         ;4800 BAUD -> TH1=#0FAH
                                        ;9600 BAUD -> TH1=#0FDH
             MOV      TCON,#11000000B   ;TF1=1 TURN TIMER 1 ON
             MOV      SCON,#01010010B   ;MODE 1 8 BIT DATA W VAR BAUD RATE

             SETB     XON               ;SET XON
             SETB     EA                ;ENABEL ALL INTERUPTS
             SETB     ES                ;ENABEL SERIAL INTERUPT
             SETB     PS                ;SERIAL INTERUPT HIGH PTIORITY

             MOV      P1,#0FFH
             JNB      P1.6,SIG
             JMP      AD
;-------------------------------------------------------------;
;       RAM TEST AND SIGNON MESAGE                            ;
;-------------------------------------------------------------;
SIG:         MOV      DPTR,#0027H
             MOVX     A,@DPTR
             CJNE     A,#55H,SIGNON
SIGNON:      MOV      DPTR,#MSG1
             ACALL    PMSG

             MOV      DPTR,#0000H
RAM_TEST:    MOV      P1,#00H
             MOV      A,#0AAH
             MOVX     @DPTR,A           ;LOAD AA HEX TO RAM
             MOVX     A,@DPTR           ;READ RAM TO ACCUMULATOR
             CJNE     A,#0AAH,RAM_TEST1 ;JMP TO ERROR IF A <> AA HEX
             MOV      A,#55H
             MOVX     @DPTR,A           ;LOAD 55 HEX TO RAM
             MOVX     A,@DPTR           ;READ RAM TO ACCUMULATOR
             CJNE     A,#55H,RAM_TEST1  ;JMP TO ERROR IF A <> 55 HEX
             INC      DPTR              ;INC RAM ADDRES
             MOV      A,DPH
             CJNE     A,#0EFH,RAM_TEST  ;LOOP IF NOT END OF RAM
             JMP      GETCM
RAM_TEST1:   MOV      R6,DPL
             MOV      R7,DPH
             MOV      DPTR,#MSG2
             ACALL    PMSG
             MOV      A,R7
             ACALL    NMOUT
             MOV      A,R6
             ACALL    NMOUT

;-------------------------------------------------------------;
;  MAIN LOOP OF THE MONITOR                                   ;
;  COMMAND RECOGNIZIG RUTINE                                  ;
;                                                             ;
;  R0.........TEMPORARY STORAGE                               ;
;  R1.........POINTER TO INPUT BUFFER                         ;
;  R2-R3......START ADDRESS FOR COMMAND                       ;
;  R4-R5......END ADDRESS FOR COMMAND                         ;
;  R6.........TEMPORARY STORAGE                               ;
;-------------------------------------------------------------;

GETCM:       ACALL    CROUT
             MOV      A,#'>'
             ACALL    ECHO              ;PROMPT OUT TO CONSOLE
             ACALL    UC_IN
             ACALL    ECHO
             CJNE     A,#'D',GETCM1
             JMP      DCMD
```

```
GETCM1:          CJNE    A,#'M',GETCM2
                 JMP     MCMD
GETCM2:          CJNE    A,#'R',GETCM3
                 JMP     RCMD
GETCM3:          CJNE    A,#'?',GETCM4
                 JMP     HCMD
GETCM4:          CJNE    A,#'G',GETCM5
                 JMP     GCMD
GETCM5:          JMP     ERROR

;-----------------------------------------------------------;
;  FUNCTION: DCMD                                           ;
;  INPUTS: NONE                                             ;
;  OUTPUTS: NONE                                            ;
;  CALLS: CROUT NMOUT ECHO HILO GETHX                       ;
;  DESCRIPTION:                                             ;
;  DISPLAY MEMORY (D) COMMAND                               ;
;-----------------------------------------------------------;
DCMD:            ACALL   GETHX
                 MOV     DPL,R2
                 MOV     DPH,R3
                 ACALL   GETHX
                 MOV     R4,02
                 MOV     R5,03
DCMD1:           PUSH    DPL
                 PUSH    DPH
                 MOV     R7,#10H
                 ACALL   CROUT
                 MOV     A,#'>'
                 ACALL   ECHO
                 MOV     A,DPH
                 ACALL   NMOUT
                 MOV     A,DPL
                 ACALL   NMOUT
DCMD2:           MOV     A,#' '
                 ACALL   ECHO
                 MOVX    A,@DPTR
                 ACALL   NMOUT
                 INC     DPTR
                 ACALL   HILO
                 JC      DCMD3
                 DJNZ    R7,DCMD2
DCMD3:           MOV     R7,#05H
                 POP     DPH
                 POP     DPL
DCMD4:           MOV     A,#' '
                 ACALL   ECHO
                 DJNZ    R7,DCMD4
                 MOV     R7,#10H
DCMD5:           MOVX    A,@DPTR
                 CJNE    A,#' ',DCMD6
                 SJMP    DCMD7
DCMD6:           JNC     DCMD7
                 MOV     A,#'.'
DCMD7:           ACALL   ECHO
                 INC     DPTR
                 ACALL   HILO
                 JC      DCMD8
                 DJNZ    R7,DCMD5
                 SJMP    DCMD1
DCMD8:           JMP     GETCM

;-----------------------------------------------------------;
;  FUNCTION:  MCMD                                          ;
;  INPUTS:  NONE                                            ;
;  OUTPUTS:  NONE                                           ;
;  CALLS:  GETCM HILO GETHX                                 ;
```

```
DESCRIPTION:
MOV DATA IN MENORY (M) COMMAND
;-----------------------------------------------------------------
MCMD:           ACALL   GETHX
                MOV     DPL,R2
                MOV     DPH,R3
:               ACALL   GETHX
                MOV     R4,02
                MOV     R5,03
                ACALL   GETHX
MCMD1:          MOVX    A,@DPTR
                MOV     R6,DPL
                MOV     R7,DPH
                MOV     DPL,R2
                MOV     DPH,R3
                MOVX    @DPTR,A
                INC     DPTR
                MOV     R2,DPL
                MOV     R3,DPH
                MOV     DPL,R6
                MOV     DPH,R7
                INC     DPTR
                ACALL   HILO
                JC      MCMD2
                SJMP    MCMD1
MCMD2:          JMP     GETCM

;-----------------------------------------------------------------;
;  FUNCTION:  RCMD                                                ;
;  INPUTS:  NONE                                                  ;
;  OUTPUTS:  NONE                                                 ;
;  CALLS: GETHX GETCM NMOUT ECHO                                  ;
;  DESCRIPTION:                                                   ;
;  SUBSTITUTE INTO MEMORY (R) COMMAND                             ;
;-----------------------------------------------------------------;
RCMD:           ACALL   GETHX
                MOV     DPL,R2
                MOV     DPH,R3
RCMD1:          CJNE    A,#' ',RCMD2
                SJMP    RCMD3
RCMD2:          CJNE    A,#',',RCMD5
RCMD3:          MOVX    A,@DPTR
                ACALL   NMOUT
                MOV     A,#'-'
                ACALL   ECHO
                ACALL   GETHX
                MOV     R6,A
                JC      RCMD4
                MOV     A,R2
                MOVX    @DPTR,A
RCMD4:          INC     DPTR
                MOV     A,R6
                SJMP    RCMD1
RCMD5:          JMP     GETCM

;-----------------------------------------------------------------;
;  FUNCTION: GCMD                                                 ;
;  INPUTS: NONE                                                   ;
;  OUTPUTS: NONE                                                  ;
;  CALLS: NONE                                                    ;
;  DESCRIPTION:                                                   ;
;  JOMP TO A PROGRAM IN THE ROM                                   ;
;-----------------------------------------------------------------;
```

```
GCMD:           ACALL   GETHX
                PUSH    02
                PUSH    03
                RET

;-----------------------------------------------------------------;
;  FUNCTION: HCMD                                                 ;
;  INPUTS: DPTR LOCATION OF MASSGE                                ;
;  OUTPUTS: NONE                                                  ;
;  CALLS: PMSG                                                    ;
;  DESCRIPTION:                                                   ;
;  SEND A HELP SCREEN TO THE CONSOLE                              ;
;-----------------------------------------------------------------;
HCMD:           MOV     DPTR,#MSG3
                ACALL   PMSG
                JMP     GETCM

;-----------------------------------------------------------------;
;  FUNCTION: PMSG                                                 ;
;  INPUTS:  DPTR LOCATION OF MASSGE                               ;
;  OUTPUTS:  NONE                                                 ;
;  CALLS:  ECHO                                                   ;
;  DESTROYS: A,DPTR                                               ;
;  DESCRIPTION:                                                   ;
;  SEND A MASSGE TO THE CONSOLE THE MASSGE END WITH FF HEX        ;
;-----------------------------------------------------------------;
PMSG:           CLR     A
                MOVC    A,@A+DPTR
                CJNE    A,#0FFH,PMSG1
                RET
PMSG1:          ACALL   ECHO
                INC     DPTR
                SJMP    PMSG
;-----------------------------------------------------------------;
;  FUNCTION:  HILO                                                ;
;  INPUTS:  DPTR 16 BIT INTEGER                                   ;
;           R4-R5 16 BIT INTEGER                                  ;
;  OUTPUTS:  CARRY - 0 IF DPTR < R4-R5                            ;
;                  - 1 IF DPTR >= R4-R5                           ;
;  CALLS:  NOTHING                                                ;
;  DESTROYS: A                                                    ;
;  DESCRIPTION:                                                   ;
;  COMPERS THE 2 16 BIT INTEGER IN DPTR AND R4-R5 THE CARRY BIT   ;
;  IS SET ACCORDING TO THE RESULT OF THE COMPARISON               ;
;-----------------------------------------------------------------;
HILO:           MOV     A,R5
                CJNE    A,DPH,HILO1
                MOV     A,R4
                CJNE    A,DPL,HILO1
                SETB    C
HILO1:          RET
;-----------------------------------------------------------------;
;  FONCTION: GETHX                                                ;
;  INPUTS: NONE                                                   ;
;  OUTPUTS: R2-R3  16 BIT INTEGER  A - THE DELIMITER              ;
;  CALLS: GETCH ECHO VALDL VALDG CNVB ERROR                       ;
;  DESCRIPTION: A,R0,R1,R2,R3,R6                                  ;
;  ACCEPTS A STRING OF HEX DIGITS FROM CONSOLE AND RETURN THEIR   ;
;  VALUE AS A 16 BIT BINARY INTEGER THE NUMBER TERMINATES WHEN    ;
;  A VALID DELIMITER IS ENCOUNTERED ILLEGAL CHARACTERS CAUSE AN   ;
;  ERROR INDICATION                                               ;
;-----------------------------------------------------------------;
GETHX:          MOV     R6,#00H
                MOV     R2,#00H
                MOV     R3,#00H
GETHX1:         ACALL   UC_IN           ;GET DIGIT FROM CONSOLE
                ACALL   ECHO            ;ECHO THE CHARACTER
                ACALL   VALDL           ;SEE IF DELIMITER
```

```
                        JNC      GETHX2            ;DELINITER ALL DONE
                        ACALL    VALDG
                        JC       GETHX4
                        ACALL    CNVBN
                        MOV      R0,A
                        MOV      A,R3
                        SWAP     A
                        ANL      A,#0F0H
                        MOV      R1,A
                        MOV      A,R2
                        SWAP     A
                        ANL      A,#0FH
                        ORL      A,R1
                        MOV      R3,A
                        MOV      A,R2
                        SWAP     A
                        ANL      A,#0F0H
                        ORL      A,R0
                        MOV      R2,A
                        INC      R6
                        SJMP     GETHX1
GETHX2:                 CJNE     R6,#00H,GETHX3
                        SETB     C
GETHX3:                 RET
GETHX4:                 JMP     ERROR

;-----------------------------------------------------------------------;
;  FUNCTION: VALDG                                                      ;
;  INPUTS: A - ASCII CHARACTER                                          ;
;  OUTPUTS: CARRY 0 IF VALID HEX DIGIT 1 IF NOT                         ;
;  CALLS: NOTHING                                                       ;
;  DESTROYS: NONE                                                       ;
;  DESCRIPTION:                                                         ;
;  RETURNS 0 IN CARRY IF ITS INPUT IS AN ASCII HEX DIGIT                ;
;-----------------------------------------------------------------------;
VALDG:                  CJNE     A,#'0',VALDG1
                        CLR      C
                        SJMP     VALDG5
VALDG1:                 JC       VALDG5
                        CJNE     A,#'9',VALDG2
                        CLR      C
                        SJMP     VALDG5
VALDG2:                 JC       VALDG4
                        CJNE     A,#'A',VALDG3
                        CLR      C
                        SJMP     VALDG5
VALDG3:                 JC       VALDG5
                        CJNE     A,#'F',VALDG4
                        CLR      C
                        SJMP     VALDG5
VALDG4:                 CPL      C
VALDG5:                 RET

;-----------------------------------------------------------------------;
;  FUNCTION: VALDL                                                      ;
;  INPUTS: A - CHARACTER                                                ;
;  OUTPUTS: CARRY 0 IF VALID DELIMITER AND 1 IF NOT                     ;
;  CALLS: NOTHING                                                       ;
;  DESTROYS: NONE                                                       ;
;  DESCRIPTION:                                                         ;
;  RETURN 0 IN CARRY IF INPUT IS VALID DELIMITER                        ;
;-----------------------------------------------------------------------;
VALDL:                  CJNE     A,#' ',VALDL1
                        CLR      C
                        SJMP     VALDL4
```

```
VALDL1:         CJNE    A,#'.',VALDL2
                CLR     C
                SJMP    VALDL4
VALDL2:         CJNE    A,#LF,VALDL3
                CLR     C
                SJMP    VALDL4
VALDL3:         SETB    C
VALDL4:         RET

;-------------------------------------------------------------
;  FUNCTION:  UC_IN
;  INPUTS:  NONE
;  OUTPUTS:  A - CHARACTER FROM CONSOL
;  CALLS:  INT_IN
;  DESTROYS:  A
;  DESCRIPTION:
;  GET A CHR FROM CONSOL AND CUNVERT LOWER CASE TO UPER CASE
;
;-------------------------------------------------------------
UC_IN:          CLR     IN
UC_IN1:         JNB     IN,UC_IN1
                CLR     IN
                MOV     A,SBUF
                CLR     ACC.7           ;MASK OFF PARITY
                CLR     C               ;IF LOWER CASE CONVERT TO
                CJNE    A,#61H,UC_IN2   ;UPER CASE
                SUBB    A,#20H
                SJMP    UC_IN4
UC_IN2:         JC      UC_IN4
                CJNE    A,#7AH,UC_IN3
                CLR     C
                SUBB    A,#20H
                SJMP    UC_IN4
UC_IN3:         JNC     UC_IN4
                CLR     C
                SUBB    A,#20H
UC_IN4:         RET

;-------------------------------------------------------------;
;  FUNCTION: INT_IN                                           ;
;  INPUTS: NONE                                               ;
;  OUTPUTS: A - CHARACTER FROM TTY                            ;
;  CALLS: NONE                                                ;
;  DESTROYS: A                                                ;
;  DESCRIPTION:                                               ;
;  INPUT CHARACTER                                            ;
;  RETURNS WITH CHR IN ACCUMULATOR                            ;
;-------------------------------------------------------------;
INT_IN:         PUSH    ACC
                MOV     A,SBUF          ;READ INPUT CHR
                CLR     RI
                SETB    IN
                POP     ACC
                RETI


;-------------------------------------------------------------;
;  FUNCTION: C_OUT                                            ;
;  INPUTS: A - CARACTER TO BE SEND TO TTY                     ;
;  OUTPUTS: NONE                                              ;
;  CALLS: NOTHING                                             ;
;  DESTROYS: A                                                ;
;  DESCRIPTION:                                               ;
;  OUTPUTS CHR IN ACCUMULATOR TO RS-232                       ;
;-------------------------------------------------------------;
```

```
C_OUT:          JNB     XON,C_OUT
                CLR     ES                      ;NO SERIAL INTERUPT
                MOV     SBUF,A
C_OUT1:         JNB     TI,C_OUT1
                CLR     TI
                SETB    ES                      ;ENABEL SERIAL INTERUPT
                RET

;-------------------------------------------------------------------;
;  FUNCTION: CNVBN                                                  ;
;  INPUTS: A - ASCII CARACTER 0-F                                   ;
;  OUTPUTS: A - 0 TO F HEX                                          ;
;  CALLS : NOTHING                                                  ;
;  DESTROYS: A                                                      ;
;  DESCRIPTION:                                                     ;
;  CONVERT THE ASCII REPRESENTATION OF A HEX NUMBER INTO ITS        ;
;  CORRESPONDING BINARY VALUE.                                      ;
;  DOSE NOT CHECK THE VALIDITY OF ITS INPUT                         ;
;-------------------------------------------------------------------;
CNVBN:          SUBB    A,#'0'
                CJNE    A,#09H,CNVBN1
                SJMP    CNVBN2
CNVBN1:         JC      CNVBN2
                SUBB    A,#07H
CNVBN2:         RET
;-------------------------------------------------------------------;
;  FUNCTION: ECHO                                                   ;
;  INPUTS: A - CARACTER TO ECHO TO CONSOLE                          ;
;  OUTPUTS: NONE                                                    ;
;  CALLS: C_OUT                                                     ;
;  DESTROYS: A                                                      ;
;  DESCRIPTION:                                                     ;
;  SENDS A CARACTER TO THE USER TERMINAL A CARRIAGE RETURN          ;
;  IS ECHOED AS CARRIAGE RETURN END LINE FEED                       ;
;-------------------------------------------------------------------;
ECHO:           ACALL   C_OUT
                CJNE    A,#CR,ECHO1
                MOV     A,#LF
                ACALL   C_OUT
ECHO1:          RET

;-------------------------------------------------------------------;
;  FUNCTION: CROUT                                                  ;
;  INPUTS: NONE                                                     ;
;  OUTPUTS: NONE                                                    ;
;  CALLS: ECHO                                                      ;
;  DESTROYS: A                                                      ;
;  DESCRIPTION:                                                     ;
;  SENDS A CARIAGE RETURN AND HENCE A LINE FEED TO THE CONSOLE      ;
;-------------------------------------------------------------------;
CROUT:          MOV     A,#CR
                ACALL   ECHO
                RET

;-------------------------------------------------------------------;
;  FUNCTION: ERROR                                                  ;
;  INPUTS: NONE                                                     ;
;  OUTPUTS: NONE                                                    ;
;  CALLS: ECHO CROUT GETCM                                          ;
;  DESTROYS: A                                                      ;
;  DESCRIPTION:                                                     ;
;  PRINT THE ERROR CARACTER ON CONSOLE FOLLWED BY CR LF             ;
;  RETURNS CONTROL TO THE COMMAND RECOGNIZER                        ;
;-------------------------------------------------------------------;

ERROR:          MOV     A,#'>'
                ACALL   ECHO
                MOV     A,#'?'
```

```
                ACALL   ECHO
                ACALL   CROUT
                MOV     SP,#STK
                JMP     GETCM

;---------------------------------------------------------------;
;  FUNCTION: PRVAL                                              ;
;  INPUTS: A - INTEGER RANGE 0 TO F HEX                         ;
;  OUTPUTS: A - ASCII CHARACTER                                 ;
;  CALLS: NOTHING                                               ;
;  DESTROYS: A                                                  ;
;  DESCRIPTION:                                                 ;
;  CONVERT A NUMBER IN THE RANGE 0 TO F HEX TO CORRESPONDING    ;
;  ASCII CHARACTER 0-9 A-F                                      ;
;---------------------------------------------------------------;
PRVAL:          PUSH    DPL
                PUSH    DPH
                MOV     DPTR,#DIGIT
                MOVC    A,@A+DPTR
                POP     DPH
                POP     DPL
                RET
DIGIT:          DB      '0123456789ABCDEF'

;---------------------------------------------------------------;
;  FUNCTION: NMOUT                                              ;
;  INPUTS: A - 8 BIT INTEGER                                    ;
;  OUTPUTS: NONE                                                ;
;  CALLS: ECHO PRVAL                                            ;
;  DESTROYS: R0                                                 ;
;  DESCRIPTION:                                                 ;
;  CONVERT THE 8 BIT INTEGER INTO 2 ASCII CHARACTER AND         ;
;  SEND THEM TO THE CONSOLE                                     ;
;---------------------------------------------------------------;
NMOUT:          PUSH    00
                MOV     R0,A
                ANL     A,#0F0H
                SWAP    A
                ACALL   PRVAL
                ACALL   ECHO
                MOV     A,R0
                ANL     A,#0FH
                ACALL   PRVAL
                ACALL   ECHO
                MOV     A,R0
                POP     00
                RET

DNMOUT:         PUSH    00
                PUSH    DPL
                PUSH    DPH
                MOV     DPTR,#DIGIT
                MOV     B,#10
                DIV     AB
                MOV     R0,A
                MOV     A,B
                MOVC    A,@A+DPTR
                MOV     D1,A
                MOV     A,R0
                MOV     B,#10
                DIV     AB
                MOV     R0,A
                MOV     A,B
                MOVC    A,@A+DPTR
                MOV     D2,A
```

```
                MOV     A,R0
                MOV     B,#10
                DIV     AB
                MOV     A,B
                MOVC    A,@A+DPTR
                CALL    C_OUT
                MOV     A,D2
                CALL    C_OUT
                MOV     A,D1
                CALL    C_OUT
                POP     DPH
                POP     DPL
                POP     00
                RET


;-------------------------------------------------------------------;
;       MASSAGES                                                    ;
;-------------------------------------------------------------------;
MSG1: DB   ODH,'**************** 8031 MONITOR ***************',ODH
      DB   ' P.C.S.I Ltd. PASCAL PESAH & ROSENWASSER DORON ',ODH,OAH
      DB   ' JET ENGINE LABORATRY TECHNION HAIFA ',ODH
      DB   ' type "?" for monitor command help ',ODH,OFFH
MSG2: DB   'RAM ERROR AT ADDRESS:  ',OFFH
MSG3: DB   ODH,'                HELP SCREEN ',ODH
      DB   'THIS HELP SCREEN:                  ?',ODH
      DB   'DISPLAY MEMORY COMMAND     D<START>.<END>',ODH
      DB   'TYPE G1100 FOR Clibration & Single Channel Reading.',ODH
      DB   'TYPE G1200 FOR Main Loop - All Channels Sampling.',ODH
      DB   'TYPE G1300 FOR Data out Via RS-232 in 16 chnnel format.',ODH
      DB   'TYPE G1400 FOR Index out Via RS-232.',ODH
      DB   'TYPE G1500 FOR data out via RS-232 in direct byte format.',ODH,

;**************************************************************************
;*                    E N D    O F    M O N I T O R
;**************************************************************************
                ORG     1000H
;-------------------------------------------------------------------;
;                                                                   ;
;       8031 CPU SINGLE BOARD COMPUTER                              ;
;       DATA ACQUISITION SYSTEM                                     ;
;       BY PESAH PASCAL & DORON ROSENWASSER                         ;
;       P.C.S.I ISRAEL                                              ;
;                                                                   ;
;-------------------------------------------------------------------;
ADC1            EQU     0F000H
ADC2            EQU     0F800H

;-------------------------------------------------------------------;
;       THIS ROUTINE DELAYS (LOC 27/100) SECONDS                    ;
;       LOCATION 27 MUST BE LOADED BY THE CALLING CODE              ;
;-------------------------------------------------------------------;

DELAY:          MOV     TMP,#00H
SELF:           DJNZ    TMP,SELF
                DJNZ    DLY,SELF
                RET

;-------------------------------------------------------------------;
;        READ THE A/D 1 RESULT                                      ;
;-------------------------------------------------------------------;
RAD1:           MOV     DPTR,#ADC1
                MOVX    A,@DPTR
                RET
```

```
;-----------------------------------------------------------------------;
;          START THE A/D 1                                              ;
;-----------------------------------------------------------------------;
SAD1:              MOV      DPTR,#ADC1
                   MOVX     @DPTR,A
                   RET

;-----------------------------------------------------------------------;
;          READ THE A/D 2 RESULT                                        ;
;-----------------------------------------------------------------------;
RAD2:              MOV      DPTR,#ADC2
                   MOVX     A,@DPTR
                   RET

;-----------------------------------------------------------------------;
;          START THE A/D 2                                              ;
;-----------------------------------------------------------------------;
SAD2:              MOV      DPTR,#ADC2
                   MOVX     @DPTR,A
                   RET

;-----------------------------------------------------------------------;
;     SELECT A CHANEL,ACCUMULATOR CONTAINS THE CHANEL NUMBER            ;
;-----------------------------------------------------------------------;
SCH:               MOV      R0,A
                   ANL      A,#00000001B
                   CJNE     A,#00000000B,SCH1
                   CLR      CH
                   SJMP     SCH2
SCH1:              SETB     CH
SCH2:              MOV      A,R0
                   RR       A
                   ANL      A,#00000111B
                   ORL      A,#11100000B
                   MOV      P1,A
                   RET

;*****************************************************************************
                   ORG      1100H
;-----------------------------------------------------------------------;
;       READ A CHANL AND DISPLAY TO CONSOLE                             ;
;-----------------------------------------------------------------------;
RCH:               MOV      DPTR,#MSG6
                   LCALL    PMSG             ;CALL MASAGE #6
                   LCALL    GETHX            ;GET CHANEL NO.(HEX)
RCH0:              MOV      A,R2
                   MOV      R7,02
                   ACALL    SCH              ;SELECT THE CHANEL
                   MOV      DLY,#05H
                   ACALL    DELAY            ;DELAY FOR 8*256*2*10^-6 mSEC
RCH1:              JB       CH,RC1
                   ACALL    SAD1             ;SELECT THE A/D TO START CONVERTION
                   SJMP     RC2
RC1:               ACALL    SAD2
RC2:               MOV      DLY,#01H
                   ACALL    DELAY            ;DELAY
                   MOV      A,R7
                   LCALL    NMOUT            ;PROMPT THE CHANEL NO.
                   MOV      A,#'-'           ;PROMPT "-" SIGN
                   LCALL    ECHO
                   JB       CH,RC3
                   ACALL    RAD1             ;READ THE A/D RESULT
                   SJMP     RC4
RC3:               ACALL    RAD2
RC4:               LCALL    NMOUT            ;PROMPT THE A/D RESULT
                   LCALL    CROUT
                   LCALL    UC IN
```

```
                CJNE    A,#' ',RCH2
                SJMP    RCH1
RCH2:           CJNE    A,#'H',RCH3
                INC     R2
                SJMP    RCH0
RCH3:           CJNE    A,#'L',RCH4
                DEC     R2
                SJMP    RCH0
RCH4:           CJNE    A,#1BH,RCH1
                LJMP    GETCM

;*************************************************************************
                ORG     1200H
;-------------------------------------------------------------;
;       SCAN ALL CHANELS INTO MEMORY                          ;
;-------------------------------------------------------------;

AD:             MOV     DPTR,#MSG7
                LCALL   PMSG
                MOV     P1,#0FFH          ;ASSIGN I/O PORT ALL TO HIGH STATE
AD1:            JB      P1.7,AD1          ;WAIT FOR LOW STATE IN I/O7 BIT
;-------------------------------------------------------------;
;       MAIN LOOP                                             ;
;-------------------------------------------------------------;

                MOV     DPTR,#MSG8
                LCALL   PMSG
                MOV     R0,#00H           ;CHANEL NO 00H
                MOV     DPTR,#0000H
                MOV     A,#01H
                MOVX    @DPTR,A
                MOV     DPTR,#0100H       ;ASSIGN MEMORY POINTER 0100H
AD2:            PUSH    DPL               ;SAVE LOW POINTER
                PUSH    DPH               ;SAVE HIGH POINTER
                MOV     A,R0              ;GET CHANEL NO.
                PUSH    00                ;SAVE CHANEL NO.
                ACALL   SCH               ;SELECT THE CHANEL
                MOV     DLY,#05H
                ACALL   DELAY             ;DELAY FOR 5*256*2*10^-6 mSEC
                JB      CH,AD21
                ACALL   SAD1              ;SELECT A/D TO START CONVERTION
                SJMP    AD22
AD21:           ACALL   SAD2
AD22:           MOV     DLY,#01H
                ACALL   DELAY             ;DELAY
                JB      CH,AD23
                ACALL   RAD1              ;READ A/D RESULT
                SJMP    AD24
AD23:           ACALL   RAD2
AD24:           POP     00                ;RESTORE CHANEL NO.
                POP     DPH               ;RESTORE HIGH POINTER
                POP     DPL               ;RESTORE LOW POINTER
                MOVX    @DPTR,A           ;PUT A/D RESULT IN MEMORY
                INC     DPTR              ;POINTER = POINTER + 1
                MOV     A,DPH             ;GET HIGH BYTE OF POINTER
                CJNE    A,#0E1H,AD3       ;IS DPH = 0E1H ?(END OF MEMORY)
                MOV     DPTR,#MSG3
                LCALL   PMSG
                LJMP    GETCM             ;YES-STOP CONVERTION & WAIT FOR COM
AD3:            INC     R0                ;NO-R0 = R0 + 1
                CJNE    R0,#10H,AD2       ;IS R0 = 010H ?(CHANEL 16)
                JNB     P1.7,AD5
                MOV     R7,DPL
                MOV     R6,DPH
                PUSH    DPL
```

13

```
                PUSH    DPH
                MOV     DPTR,#0000H
                MOVX    A,@DPTR
                MOV     DPL,A
                MOV     A,R6
                MOVX    @DPTR,A
                MOV     A,R7
                INC     DPTR
                MOVX    @DPTR,A
                INC     DPTR
                MOV     A,DPL
                MOV     DPTR,#0000H
                MOVX    @DPTR,A
                POP     DPH
                POP     DPL
AD4:            JB      P1.7,AD4
AD5:            MOV     R0,#00H             ;YES-SET R0 = 00H
                SJMP    AD2                 ;NO-GOTO MAIN LOOP AGAIN

;**********************************************************************
                ORG     1300H
;-------------------------------------------------------------------;
;               OUTPUT SETS OF 16 HEX BYTES FROM RAM                ;
;-------------------------------------------------------------------;


                MOV     DPTR,#0100H
SOUT1:          MOV     R7,#10H
                LCALL   CROUT
SOUT2:          MOVX    A,@DPTR
                LCALL   NMOUT
                MOV     A,#' '
                CALL    C_OUT
                INC     DPTR
                MOV     A,DPH
                CJNE    A,#0E1H,SOUT3
                MOV     DPTR,#MSG9
                LCALL   PMSG
                MOV     DPTR,#MSG3
                LCALL   PMSG
                LJMP    GETCM
SOUT3:          DJNZ    R7,SOUT2
                SJMP    SOUT1

;**********************************************************************
                ORG 1400H
;-------------------------------------------------------------------;
;               OUTPUT INDEX TO PC via RS-232                       ;
;-------------------------------------------------------------------;
                MOV     DPTR,#0000H
                MOV     R7,#07FH
ADD1:           LCALL   CROUT
                INC     DPTR
                MOVX    A,@DPTR
                LCALL   NMOUT
                INC     DPL
                MOVX    A,@DPTR
                LCALL   NMOUT
                DJNZ    R7,ADD1

                LCALL   CROUT
                MOV     DPTR,#MSG10
                LCALL   PMSG
                MOV     DPTR,#0000H
                MOVX    A,@DPTR
```

```
                DEC     A
                RR      A
                ANL     A,#07FH
                LCALL   NMOUT
                MOV     A,#'H'
                CALL    C_OUT
                LCALL   CROUT
                MOV     DPTR,#MSG9
                LCALL   PMSG
                MOV     DPTR,#MSG3
                LCALL   PMSG
                LJMP    GETCM

;***************************************************************************
                ORG 1500H
;--------------------------------------------------------------------------;
;               OUTPUT DATA TO PC via RS-232 CHR FORMAT                    ;
;--------------------------------------------------------------------------;
                MOV     DPTR,#0100H
COUT1:          MOVX    A,@DPTR
                LCALL   C_OUT
                INC     DPTR
                MOV     A,DPH
                CJNE    A,#0E1H,COUT1
                MOV     DPTR,#MSG9
                LCALL   PMSG
                MOV     DPTR,#MSG3
                LCALL   PMSG
                LJMP    GETCM

;--------------------------------------------------------------------------;
;                               M A S A G E S  !!                          ;
;--------------------------------------------------------------------------;

MSG6:           DB      'ENTER CHANEL NUMBER TO READ:  ',0FFH
MSG7:           DB      0DH,'WAITING FOR RECORD COMMAND...',0DH,0DH,0FFH
MSG8:           DB      'PLEASE WAIT RECORDING CHANELS...',0DH,0FFH
MSG9:           DB      0DH,'DOWNLOADING IS FINISHED!!!',0DH,0DH,0DH,0FFH
MSG10:          DB      0DH,'THE NUMBER OF SESIONS IS: ',0DH,0FFH
                END
```

```
                    END$ROM(LARGE)
$CODE
$debug
/**************************************************************************************/
/*                 TELEMETRIC RECORDER P.C.S.I HAIFA ISRAEL.                          */
/*                 BY PASCAL PESAH & ASHER EFRATY                                      */
/**************************************************************************************/
MAIN$MIN:
DO;
$INCLUDE (REG51.DCL)
DECLARE
CH1  BIT AT(90H) REG,
CH2  BIT AT(91H) REG,
CH3  BIT AT(92H) REG,
CH4  BIT AT(93H) REG,
CH5  BIT AT(94H) REG,
CH6  BIT AT(95H) REG,
CH7  BIT AT(96H) REG;
DECLARE DCL LITERALLY 'DECLARE';
DECLARE AUX LITERALLY 'AUXILIARY';
DECLARE PROC LITERALLY 'PROCEDURE';

/**************************************************************************************/

DCL (pointer,idx) WORD;
DCL data_out BASED pointer BYTE AUX;
DCL section BASED idx WORD AUX;
DCL data_in BYTE;                              /* The legth of the interval. */
DCL i BYTE;                                    /* Temperary counter.         */
DCL chr BYTE;                                  /* Chr byte.                  */
DCL new_ope BIT;                               /* Indicate present operation */
DCL old_ope BIT;                               /* Indicate last operation    */
DCL RE BIT;                                    /* Record flag                */
DCL s1 BIT;
DCL ix     literally 't0' ;                    /* Index  Pushbutton.  t0     */
DCL dump   literally 't1';                     /* Dump   Pushbutton.  t1     */
DCL record literally 'int1';                   /* Record Switch.  int1       */

/************************************************************************************/


SET$BAUD$RATE:                                    /* INIT RS232 AND SET BAUD RATE */
PROC (N);
DCL N BYTE;
TMOD=29H;
DO CASE N;
        TH1=040H;                                 /*        150  BAUD          */
        TH1=0A0H;                                 /*        300  BAUD          */
        TH1=0D0H;                                 /*        600  BAUD          */
        TH1=0E8H;                                 /*       1200  BAUD          */
        TH1=0F4H;                                 /*       2400  BAUD          */
        TH1=0FAH;                                 /*       4800  BAUD          */
        TH1=0FDH;                                 /*       9600  BAUD          */
END;
TCON=0C0H;
SCON=72H;
EA=0;
ES=0;
PS=1;
END SET$BAUD$RATE;

PUT$CHR: PROC(CHAR);                              /*    PRINT A CHAR TO RS232     */
```

```
  DO WHILE NOT TI;
  END;
  TI=0;
  SBUF=CHAR;
END PUT$CHR;

GET$CHR: PROC BYTE;                          /* GET A CHAR FROM RS232 AND ECHO */
   DCL CHR BYTE;
   DO WHILE NOT RI;
   END;
   RI=0;
   CHR=SBUF;
   CALL PUT$CHR(CHR);
   RETURN(CHR);
END GET$CHR;

PMSG: PROC (ADD);                              /* PRINT MASAGE TO RS-232        */
  DCL ADD WORD;
  DCL CHR BASED ADD BYTE CONSTANT;
  DO WHILE CHR <>0FFH;
        CALL PUT$CHR(CHR);
        ADD=ADD+1;
  END;
END PMSG;

  DCL MSG1(*) BYTE CONSTANT
         ('.      TECHNION - TURBO & ENGINE LABORATRY   '
,0DH,0AH, '*******   RADIO CONTROL RECORDER  V1.0 *******',0DH,0AH,0FFH );
  DCL·MSG2(*) BYTE CONSTANT
         ('START DUMPING RECORDS...',0DH,0AH,0FFH ):
  DCL MSG3(*) BYTE CONSTANT
         ('ALL RECORDS DUMPED SUCCESSFULY!!!',0DH,0AH,
          'PRESS INDEX BUTTON FOR SESSION INDEX',0DH,0AH,0FFH );
  DCL MSG4(*) BYTE CONSTANT
         ('DUMPING SESSION INDEX...',0DH,0AH,0FFH );
  DCL MSG5(*) BYTE CONSTANT
         ('SESSION INDEX DUMPED SUCCESSFULY!!!',0DH,0AH,0FFH );

PROMPT:PROC;                                   /* PROMPT '>' VIA RS 232         */
  DCL CHR BYTE;
  CHR=0DH;
  CALL PUT$CHR(CHR);
  CHR=0AH;
  CALL PUT$CHR(CHR);
  CHR='>';
  CALL PUT$CHR(CHR);
END PROMPT;

NUMOUT: PROCEDURE(VALUE,BASE,WIDTH);          /* PRINT NUMBER OUT TO CONSOL    */
    DCL VALUE ADDRESS;
    DCL (BASE,WIDTH,I) BYTE;
    DCL DIGITS(*) BYTE CONSTANT ('0123456789ABCDEF');
    DCL DD(5) BYTE;
    DO I=1 TO WIDTH;
        DD(I)=(DIGITS(VALUE MOD BASE));
        VALUE=VALUE / BASE;
    END;
    DO I=0 TO (WIDTH-1);
        CALL PUT$CHR(DD(WIDTH-I));
    END;
END NUMOUT;
```

```
/*                        WRITE DATA TO MEMORY PROCEDURE                      */
/****************************************************************************/

write_data: PROC;
  data_out=data_in;              /* data_out is based on pointer              */
  IF pointer < 0E100H THEN       /* checks if not end of memory               */
     pointer=pointer+1;          /* Increment address of the data_out pointer */
END write_data;

/****************************************************************************/
/*              READS DATA FROM RADIO SIGNALS AND STORE TO MEMORY           */
/****************************************************************************/

read_data: PROC;
  POINTER=POINTER+1;
  DO WHILE CH1=1;
    data_in=data_in+1;
    i=i*1;
  END;
  DATA_OUT=DATA_IN;
  data_in=0;
  POINTER=POINTER-1;
  i=1;
  DO WHILE i<=6;                 /* There are 7 traces in evry pattern        */
    IF CH2=1 THEN DO;
        DO WHILE CH2=1;
            DATA_IN=DATA_IN+1;
            I=I*1;
        END;
        POINTER=POINTER+2;
        DATA_OUT=DATA_IN;
        POINTER=POINTER-2;
        DATA_IN=0;
        I=I+1;
    END;
    IF CH3=1 THEN DO;
        DO WHILE CH3=1;
            DATA_IN=DATA_IN+1;
            I=I*1;
        END;
        POINTER=POINTER+3;
        DATA_OUT=DATA_IN;
        POINTER=POINTER-3;
        DATA_IN=0;
        I=I+1;
    END;
    IF CH4=1 THEN DO;
        DO WHILE CH4=1;
            DATA_IN=DATA_IN+1;
            I=I*1;
        END;
        POINTER=POINTER+4;
        DATA_OUT=DATA_IN;
        POINTER=POINTER-4;
        DATA_IN=0;
        I=I+1;
    END;
    IF CH5=1 THEN DO;
        DO WHILE CH5=1;
            DATA_IN=DATA_IN+1;
            I=I*1;
        END;
        POINTER=POINTER+5;
        DATA_OUT=DATA_IN;
        POINTER=POINTER-5;
```

```
        I=I+1.
        ~~old_ope=new_ope;~~
        IF data_in > 100 THEN
             new_ope=1;
        ELSE
             new_ope=0;
        data_in=0;
    END;
    IF CH6=1 THEN DO;
        DO WHILE CH6=1;
            DATA_IN=DATA_IN+1;
            I=I*1;
        END;
        POINTER=POINTER+6;
        DATA_OUT=DATA_IN;
        POINTER=POINTER-6;
        DATA_IN=0;
        I=I+1;
    END;
    IF CH7=1 THEN DO;
        DO WHILE CH7=1;
            DATA_IN=DATA_IN+1;
            I=I*1;
        END;
        POINTER=POINTER+7;
        DATA_OUT=DATA_IN;
        POINTER=POINTER-7;
        DATA_IN=0;
        I=I+1;
    END;
  END;
  POINTER=POINTER+7,
  IF old_ope=0 AND new_ope=1 THEN
       DO;
             section=pointer-7;
             if idx<97 then
             idx=idx+2;
       END;
  IF new_ope=0 THEN
       pointer=pointer-7;
END read_data;

/****************************************************************************/
/*         INIT ALL MEMORY,RS-232,FLAGS,idx,POINTER PROCEDURE              */
/****************************************************************************/

init$all: PROC;
   CALL set$baud$rate(5);
   CALL pmsg(.msg1(0));
   CALL prompt;                /* PROMPT '>' VIA RS-232        */
   new_ope=0;
   old_ope=0;
   s1=1;
   DO idx=0 TO 100 BY 2;
      section=256;
   END;
   idx=0;
   pointer=100H;
   t0=1;
   t1=1;
   int1=1;
END init$all;


dump_:PROC;
DCL (num) WORD;
```

```
CALL PMSG(.MSG2(0));
CALL PROMPT;
num=pointer;
pointer=100H;
DO WHILE num > pointer;
    CALL put$chr(data_out);    /* numout(data_out,10,3)    */
    pointer=pointer+1;
END;
CALL PMSG(.MSG3(0));
CALL PROMPT;
END DUMP_;


IX_:PROC;
DCL (DD) WORD;
CALL PMSG(.MSG4(0));
CALL PROMPT;
DO idx=0 TO 98 BY 2;
    DD =(section-256)/7;
    CALL numout(dd,10,5);
    CALL put$chr(0DH);
    CALL put$chr(0AH);
END;
CALL PMSG(.MSG5(0));
CALL PROMPT;
CALL PMSG(.MSG1(0));
CALL PROMPT;
END IX_;

/****************************************************************************
/*     RECORD A CHNNEL PROCEDURE                                               *
/****************************************************************************
rec: PROC;
DCL s  BYTE;
  DO WHILE RE AND RECORD,
       chr=' ';
       DO WHILE CH1=0;              /* Detect for CH1 pulse. */
       END;
       IF s1 THEN CALL read_data;   /* if CH1 pulse & s1=1 read */
       s1=NOT s1;
       IF POINTER=0E100H THEN RE=0;
  END;
END rec;


/*************************************************************************/

DCL init LABEL PUBLIC;
init:
RE=1;
P1=0FFH;
CALL init$all;
DO WHILE 1;
    IF record AND RE THEN CALL rec;
    IF NOT ix THEN CALL ix_;
    IF NOT dump THEN CALL dump_;
END;
END;
```

6C

Binary to Digital Converter

Generated by

Diego Crupnicoff

December 15 to December 20,1991

for

TLIT/JPL - USAF Projects

Source Code

Program starts at the last
four pages

The first pages
include "FILE"

This program has been ordered for the
3rd-generation flight recording and
ground computers

```
#ifndef _SIZE_T DEFINED
typedef unsigned int size_t;
#define _SIZE_T_DEFINED
#endif

#ifndef NO_EXT_KEYS     /* extensions enabled */
        #define _CDECL  cdecl
#else /* extensions not enabled */
        #define _CDECL
#endif /* NO_EXT_KEYS */


/* function prototypes */

void * _CDECL memccpy(void *, void *, int, unsigned int);
void * _CDECL memchr(const void *, int, size_t);
int _CDECL memcmp(const void *, const void *, size_t);
void * _CDECL memcpy(void *, const void *, size_t);
int _CDECL memicmp(void *, void *, unsigned int);
void * _CDECL memset(void *, int, size_t);
void _CDECL movedata(unsigned int, unsigned int, unsigned int, unsigned int, unsigned int);
```

```
#ifndef NO_EXT_KEYS     /* extensions enabled */
        #define _CDECL  cdecl
        #define _NEAR   near
#else /* extensions not enabled */
        #define _CDECL
        #define _NEAR
#endif /* NO_EXT_KEYS */

/*
 * This declaration allows the user access to the ctype look-up
 * array _ctype defined in ctype.obj by simply including ctype.h
 */

extern unsigned char _NEAR _CDECL _ctype[];

/* set bit masks for the possible character types */

#define _UPPER          0x1     /* upper case letter */
#define _LOWER          0x2     /* lower case letter */
#define _DIGIT          0x4     /* digit[0-9] */
#define _SPACE          0x8     /* tab, carriage return, newline, */
                                /* vertical tab or form feed */
#define _PUNCT          0x10    /* punctuation character */
#define _CONTROL        0x20    /* control character */
#define _BLANK          0x40    /* space char */
#define _HEX            0x80    /* hexadecimal digit */

/* the character classification macro definitions */

#define isalpha(c)              ( (_ctype+1)[c] & (_UPPER|_LOWER) )
#define isupper(c)              ( (_ctype+1)[c] & _UPPER )
#define islower(c)              ( (_ctype+1)[c] & _LOWER )
#define isdigit(c)              ( (_ctype+1)[c] & _DIGIT )
#define isxdigit(c)             ( (_ctype+1)[c] & _HEX )
#define isspace(c)              ( (_ctype+1)[c] & _SPACE )
#define ispunct(c)              ( (_ctype+1)[c] & _PUNCT )
#define isalnum(c)              ( (_ctype+1)[c] & (_UPPER|_LOWER|_DIGIT) )
#define isprint(c)              ( (_ctype+1)[c] & (_BLANK|_PUNCT|_UPPER|_LOWER|_DIGIT) )
#define isgraph(c)              ( (_ctype+1)[c] & (_PUNCT|_UPPER|_LOWER|_DIGIT) )
#define iscntrl(c)              ( (_ctype+1)[c] & _CONTROL )

#define toupper(c)              ( (islower(c)) ? _toupper(c) : (c) )
#define tolower(c)              ( (isupper(c)) ? _tolower(c) : (c) )

#define _tolower(c)             ( (c)-'A'+'a' )
#define _toupper(c)             ( (c)-'a'+'A' )

#define isascii(c)              ( (unsigned)(c) < 0x80 )
#define toascii(c)              ( (c) & 0x7f )

/* MS C version 2.0 extended ctype macros */

#define iscsymf(c)              (isalpha(c) || ((c) == '_'))
#define iscsym(c)               (isalnum(c) || ((c) == '_'))
```

```
#ifndef _SIZE_T_DEFINED
typedef unsigned int size_t;
#define _SIZE_T_DEFINED
#endif

#ifndef NO_EXT_KEYS     /* extensions enabled */
        #define _CDECL  cdecl
        #define _NEAR   near
#else /* extensions not enabled */
        #define _CDECL
        #define _NEAR
#endif /* NO_EXT_KEYS */


/* definition of the return type for the onexit() function */

#define EXIT_SUCCESS    0
#define EXIT_FAILURE    1

#ifndef _ONEXIT_T_DEFINED
typedef int (_CDECL * _CDECL onexit_t)();
#define _ONEXIT_T_DEFINED
#endif


/* Data structure definitions for div and ldiv runtimes. */

#ifndef _DIV_T_DEFINED

typedef struct {
        int quot;
        int rem;
} div_t;

typedef struct {
        long quot;
        long rem;
} ldiv_t;

#define _DIV_T_DEFINED
#endif

/* Maximum value that can be returned by the rand function. */

#define RAND_MAX 0x7fff


/* min and max macros */

#define max(a,b)        (((a) > (b)) ? (a) : (b))
#define min(a,b)        (((a) < (b)) ? (a) : (b))


/* sizes for buffers used by the _makepath() and _splitpath() functions.
```

```
 * note that the sizes include space for 0-terminator
 */

#define _MAX_PATH      144          /* max. length of full pathname */
#define _MAX_DRIVE     3            /* max. length of drive component */
#define _MAX_DIR       130          /* max. length of path component */
#define _MAX_FNAME     9            /* max. length of file name component */
#define _MAX_EXT       5            /* max. length of extension component */

/* external variable declarations */

extern int _NEAR _CDECL errno;                  /* XENIX style error number */
extern int _NEAR _CDECL _doserrno;              /* MS-DOS system error value */
extern char * _NEAR _CDECL sys_errlist[];       /* perror error message table */
extern int _NEAR _CDECL sys_nerr;               /* # of entries in sys_errlist table */

extern char ** _NEAR _CDECL environ;            /* pointer to environment table */

extern unsigned int _NEAR _CDECL _psp;          /* Program Segment Prefix */

extern int _NEAR _CDECL _fmode;                 /* default file translation mode */

/* DOS major/minor version numbers */

extern unsigned char _NEAR _CDECL _osmajor;
extern unsigned char _NEAR _CDECL _osminor;

#define DOS_MODE       0       /* Real Address Mode */
#define OS2_MODE       1       /* Protected Address Mode */

extern unsigned char _NEAR _CDECL _osmode;


/* function prototypes */

double _CDECL atof(const char *);
double _CDECL strtod(const char *, char * *);
ldiv_t _CDECL ldiv(long, long);

void   _CDECL abort(void);
int    _CDECL abs(int);
int    _CDECL atexit(void (_CDECL *)(void));
int    _CDECL atoi(const char *);
long   _CDECL atol(const char *);
void * _CDECL bsearch(const void *, const void *, size_t, size_t, int (_CDECL *)'
void * _CDECL calloc(size_t, size_t);
div_t  _CDECL div(int, int);
char * _CDECL ecvt(double, int, int *, int *);
void   _CDECL exit(int);
void   _CDECL _exit(int);
char * _CDECL fcvt(double, int, int *, int *);
void   _CDECL free(void *);
char * _CDECL gcvt(double, int, char *);
char * _CDECL getenv(const char *);
char * _CDECL itoa(int, char *, int);
long   _CDECL labs(long);
unsigned long _CDECL _lrotl(unsigned long, int);
unsigned long _CDECL _lrotr(unsigned long, int);
char * _CDECL ltoa(long, char *, int);
void   _CDECL _makepath(char *, char *, char *, char *, char *);
void * _CDECL malloc(size_t);
```

```
onexit_t _CDECL onexit(onexit_t);
void    _CDECL perror(const char *);
int     _CDECL putenv(char *);
void    _CDECL qsort(void *, size_t, size_t, int (_CDECL *)(const void *, const void *));
unsigned int _CDECL _rotl(unsigned int, int);
unsigned int _CDECL _rotr(unsigned int, int);
int     _CDECL rand(void);
void *  _CDECL realloc(void *, size_t);
void    _CDECL _searchenv(char *, char *, char *);
void    _CDECL _splitpath(char *, char *, char *, char *, char *);
void    _CDECL srand(unsigned int);
long    _CDECL strtol(const char *, char * *, int);
unsigned long _CDECL strtoul(const char *, char * *, int);
void    _CDECL swab(char *, char *, int);
int     _CDECL system(const char *);
char *  _CDECL ultoa(unsigned long, char *, int);

#ifndef tolower                 /* tolower has been undefined - use function */
int _CDECL tolower(int);
#endif  /* tolower */

#ifndef toupper                 /* toupper has been undefined - use function */
int     _CDECL toupper(int);
#endif  /* toupper */



#ifndef _INO_T_DEFINED
typedef unsigned short ino_t;           /* i-node number (not used on DOS) */
#define _INO_T_DEFINED
#endif

#ifndef _TIME_T_DEFINED
typedef long time_t;
#define _TIME_T_DEFINED
#endif

#ifndef _DEV_T_DEFINED
typedef short dev_t;                    /* device code */
#define _DEV_T_DEFINED
#endif

#ifndef _OFF_T_DEFINED
typedef long off_t;                     /* file offset value */
#define _OFF_T_DEFINED
#endif



#ifndef NO_EXT_KEYS     /* extensions enabled */
        #define CDECL   cdecl
#else                   /* extensions not enabled */
        #define CDECL
#endif /* NO_EXT_KEYS */

#ifndef _TIME_T_DEFINED
typedef long time_t;
#define _TIME_T_DEFINED
#endif
```

```
#ifndef _STAT_DEFINED
struct stat {
        dev_t st_dev;
        ino_t st_ino;
        unsigned short st_mode;
        short st_nlink;
        short st_uid;
        short st_gid;
        dev_t st_rdev;
        off_t st_size;
        time_t st_atime;
        time_t st_mtime;
        time_t st_ctime;
        };
#define _STAT_DEFINED
#endif

#define S_IFMT          0170000         /* file type mask */
#define S_IFDIR         0040000         /* directory */
#define S_IFCHR         0020000         /* character special */
#define S_IFREG         0100000         /* regular */
#define S_IREAD         0000400         /* read permission, owner */
#define S_IWRITE        0000200         /* write permission, owner */
#define S_IEXEC         0000100         /* execute/search permission, owner */


/* function prototypes */

int CDECL fstat(int, struct stat *);
int CDECL stat(char *, struct stat *);
```

```
#ifndef NO_EXT_KEYS    /* extensions enabled */
        #define _CDECL  cdecl
#else /* extensions not enabled */
        #define _CDECL
#endif /* NO_EXT_KEYS */

/* function prototypes */

int _CDECL access(char *, int);
int _CDECL chmod(char *, int);
int _CDECL chsize(int, long);
int _CDECL close(int);
int _CDECL creat(char *, int);
int _CDECL dup(int);
int _CDECL dup2(int, int);
int _CDECL eof(int);
long _CDECL filelength(int);
int _CDECL isatty(int);
int _CDECL locking(int, int, long);
long _CDECL lseek(int, long, int);
char * _CDECL mktemp(char *);
int _CDECL open(char *, int, ...);
int _CDECL read(int, char *, unsigned int);
int _CDECL remove(const char *);
int _CDECL rename(const char *, const char *);
int _CDECL setmode(int, int);
int _CDECL sopen(char *, int, int, ...);
long _CDECL tell(int);
int _CDECL umask(int);
int _CDECL unlink(const char *);
int _CDECL write(int, char *, unsigned int);
#ifndef NO_EXT_KEYS    /* extensions enabled */
        #define _CDECL  cdecl
#else /* extensions not enabled */
        #define _CDECL
#endif /* NO_EXT_KEYS */

/* function prototypes */

char * _CDECL cgets(char *);
int _CDECL cprintf(char *, ...);
int _CDECL cputs(char *);
int _CDECL cscanf(char *, ...);
int _CDECL getch(void);
int _CDECL getche(void);
int _CDECL inp(unsigned int);
unsigned _CDECL inpw(unsigned int);
int _CDECL kbhit(void);
int _CDECL outp(unsigned int, int);
unsigned _CDECL outpw(unsigned int, unsigned int);
int _CDECL putch(int);
int _CDECL ungetch(int);


#ifndef _SIZE_T_DEFINED
typedef unsigned int size_t;
#define _SIZE_T_DEFINED
#endif

#ifndef _VA_LIST_DEFINED
```

```
#ifndef NO_EXT_KEYS     /* extensions enabled */
        #define _CDECL  cdecl
        #define _NEAR   near
#else /* extensions not enabled */
        #define _CDECL
        #define _NEAR
#endif /* NO_EXT_KEYS */


/* buffered I/O macros */

#define  BUFSIZ 512
#define  _NFILE 20
#define  EOF    (-1)

#ifndef _FILE_DEFINED
#define  FILE   struct _iobuf
#define  _FILE_DEFINED
#endif

/* P_tmpnam: Directory where temporary files may be created.
 * L_tmpnam size =  size of P_tmpdir
 *              + 1 (in case P_tmpdir does not end in "\\")
 *              + 6 (for the temp number string)
 *              + 1 (for the null terminator)
 */

#define  P_tmpdir "\\"
#define  L_tmpnam sizeof(P_tmpdir)+8

#define  SEEK_CUR 1
#define  SEEK_END 2
#define  SEEK_SET 0

#define  FILENAME_MAX 63
#define  FOPEN_MAX 20
#define  SYS_OPEN 20
#define  TMP_MAX 32767


/* define NULL pointer value */

#if (defined(M_I86SM) || defined(M_I86MM))
#define  NULL   0
#elif (defined(M_I86CM) || defined(M_I86LM) || defined(M_I86HM))
#define  NULL   0L
#endif


/* define file control block */

#ifndef _IOB_DEFINED
extern FILE {
        char *_ptr;
        int   _cnt;
        char *_base;
        char  _flag;
        char  _file;
        } _NEAR _CDECL _iob[];
#define _IOB_DEFINED
#endif
```

```
#define  stdin  (&_iob[0])
#define  stdout (&_iob[1])
#define  stderr (&_iob[2])
#define  stdaux (&_iob[3])
#define  stdprn (&_iob[4])

#define  _IOREAD       0x01
#define  _IOWRT        0x02

#define  _IOFBF        0x0
#define  _IOLBF        0x40
#define  _IONBF        0x04

#define  _IOMYBUF      0x08
#define  _IOEOF        0x10
#define  _IOERR        0x20
#define  _IOSTRG       0x40
#define  _IORW         0x80

#define getc(f)        (--(f)->_cnt >= 0 ? 0xff & *(f)->_ptr++ : _filbuf(f))
#define putc(c,f)      (--(f)->_cnt >= 0 ? 0xff & (*(f)->_ptr++ = (char)(c)) \
                            :  _flsbuf((c),(f)))
#define getchar()      getc(stdin)
#define putchar(c)     putc((c),stdout)

#define feof(f)        ((f)->_flag & _IOEOF)
#define ferror(f)      ((f)->_flag & _IOERR)
#define fileno(f)      ((int)(unsigned char)(f)->_file)

/* function prototypes */

int _CDECL _filbuf(FILE *);
int _CDECL _flsbuf(int, FILE *);
void _CDECL clearerr(FILE *);
int _CDECL fclose(FILE *);
int _CDECL fcloseall(void);
FILE * _CDECL fdopen(int, char *);
int _CDECL fflush(FILE *);
int _CDECL fgetc(FILE *);
int _CDECL fgetchar(void);
int _CDECL fgetpos(FILE *, fpos_t *);
char * _CDECL fgets(char *, int, FILE *);
int _CDECL flushall(void);
FILE * _CDECL fopen(const char *, const char *);
int _CDECL fprintf(FILE *, const char *, ...);
int _CDECL fputc(int, FILE *);
int _CDECL fputchar(int);
int _CDECL fputs(const char *, FILE *);
size_t _CDECL fread(void *, size_t, size_t, FILE *);
FILE * _CDECL freopen(const char *, const char *, FILE *);
int _CDECL fscanf(FILE *, const char *, ...);
int _CDECL fsetpos(FILE *, const fpos_t *);
int _CDECL fseek(FILE *, long, int);
long _CDECL ftell(FILE *);
size_t _CDECL fwrite(const void *, size_t, size_t, FILE *);
char * _CDECL gets(char *);
int _CDECL getw(FILE *);
void _CDECL perror(const char *);
int _CDECL printf(const char *, ...);
int _CDECL puts(const char *);
```

```
int _CDECL putw(int, FILE *);
int _CDECL remove(const char *);
int _CDECL rename(const char *, const char *);
void _CDECL rewind(FILE *);
int _CDECL rmtmp(void);
int _CDECL scanf(const char *, ...);
void _CDECL setbuf(FILE *, char *);
int _CDECL setvbuf(FILE *, char *, int, size_t);
int _CDECL sprintf(char *, const char *, ...);
int _CDECL sscanf(const char *, const char *, ...);
char * _CDECL tempnam(char *, char *);
FILE * _CDECL tmpfile(void);
char * _CDECL tmpnam(char *);
int _CDECL ungetc(int, FILE *);
int _CDECL unlink(const char *);
int _CDECL vfprintf(FILE *, const char *, va_list);
int _CDECL vprintf(const char *, va_list);
int _CDECL vsprintf(char *, const char *, va_list);



#include <stdio.h>
#include <io.h>
#include <conio.h>
#include <stdlib.h>
#include <memory.h>
#include <ctype.h>
#include <sys/types.h>
#include <sys/stat.h>


#define HT      0x09
#define CR      0x0D
#define LF      0x0A
#define FALSE   0
#define TRUE    1

typedef int             BOOL;
typedef unsigned char   BYTE;
typedef unsigned int    WORD;
typedef unsigned long   DWORD;


char * mapping[256] = { "000","001","002","003","004","005","006","007","008","009", ...255

BOOL translate_file( FILE *, FILE *, long,BOOL);

void main( int argc, char *argv[] )
        {
        struct stat FileStat;
        FILE *fdInput;
        FILE *fdOutput;
        int  c;
        BOOL bHalf=FALSE;

        if ((argc != 3)&&(argc !=4))
                {
                fprintf( stderr, "Usage: Bin2Dec <infile> <outfile> [-H]\r\n" );
                exit( 1 );
```

```
		}

	if( argc == 4 )
		{
		if( ((argv[3][0] == '/') || (argv[3][0] == '-')) && (toupper( argv[3][1] ) == 'H') )
			bHalf = TRUE;
		else
			{
			fprintf( stderr, "Bin2Dec: invalid option '%s'\n",argv[3] );
			exit( 1 );
			}
		}

	/* Try to open the input and output files in binary mode. */
	if( (fdInput = fopen( argv[1], "rb" )) == NULL )
		{
		fprintf( stderr, "Bin2Dec: can't open '%s'\n", argv[1] );
		exit( 1 );
		}
	stat(argv[1],&FileStat);
	printf( "Source file size: %ul \r\n" ,FileStat.st_size);


	if( !access( argv[2], 0 ) )
    {
		printf( "Destination file %s exists. Overwrite ? (s/n) \r\n" ,argv[2]);
		c = getch();
		if( c != 'y' && c != 'Y' )
			exit( 1 );
		printf( "\n" );
    }

	if( (fdOutput = fopen( argv[2], "wb" )) == NULL )
		{
		fclose( fdInput );
		fprintf( stderr, "Bin2Dec: can't open '%s'\n", argv[2] );
		exit( 1 );
		}

    /* Translate the input file to the output file. */
	if (translate_file( fdInput, fdOutput ,FileStat.st_size,bHalf))
		printf("%s succesfully converted to %s \r\n",argv[1],argv[2]);
	else
		printf("unexpected end of file in %s \r\n",argv[1]);

	/* Close the files and exit. */
	fclose( fdInput );
	fclose( fdOutput );
	exit( 0 );
	}


BOOL translate_file( FILE *fdIn, FILE *fdOut ,long nFileSize, BOOL bHalf)
	{
	long nBytes;
	unsigned char c;
```

```
int nChan;
BOOL bSucc;

nBytes = 0;
bSucc = FALSE;
while((nBytes<nFileSize)&&(!bSucc))
        {
        while((nBytes++<nFileSize)&&(getc(fdIn)!='G'));
        if((nBytes++<nFileSize)&&(getc(fdIn)=='1'))
                if((nBytes++<nFileSize)&&(getc(fdIn)=='5'))
                        if((nBytes++<nFileSize)&&(getc(fdIn)=='0'))
                                if((nBytes++<nFileSize)&&(getc(fdIn)=='0'))
                                        if((nBytes++<nFileSize)&&(getc(fdIn)==CR))
                                                if((nBytes++<nFileSize)&&(getc(fdIn)==LF))
                                                        bSucc = TRUE;
        }

if (bSucc)
        {
        while(nBytes<nFileSize)
                {
                for(nChan=0;nChan<16;nChan++,nBytes++)
                        {
                        c = getc(fdIn);
                        fputs(mapping[c],fdOut);
                        putc(HT,fdOut);
                        printf(mapping[c]);
                        printf(" ");
                        }
                if (bHalf) for(nChan=0;nChan<16;nChan++,nBytes++,getc(fdIn)); //discards one line
                fputs("\r\n",fdOut);
                printf("\r\n");
                }
        }

return(bSucc);
}
```

<u>Appendix B</u>

The Following

<u>Programs</u> were

<u>Developed for the new 3rd generation</u>

<u>flight vectoring and ground computers</u>

Data downloading procedure:

1. Connecting to the plane: After the landing of the plane, and with the laptop computer switched off, connect the the serial cable from the plane to the computer.

2. Switch the computer on. and wait for the booting procedure to be completed.

3. Run the communication software (PROCOMM)

4. Setting the baud rate: Once inside the Procomm, set the baud rate to 4800 or 9600 bauds acording to the Eprom installed in the plane. To set the baud rate, use *ALT-P* and follow the on screen instructions. Choose: no parity, 8 bits, 1 stop bit and the corresponding baud rate (4800 or 9600).

5. Checking the communication: Type *'?'* and you should get the help screen of the plane computer. If not, check the cables and the communication settings and try again.

6. Activing the translate table: Type *ALT-W* and then make the table active with *F3* Type *'0' Enter '1' Enter,* in order to get all the incomming zeros tarnslated to ones. (This last operation is not recquired, if after activing the tarnslation table with F3 the '0' is already set to be translated to '1')

7. Downloading the data: Press the *PgDn* key and select the option number *7*(ASCII) from the downloading menu. The program will prompt you for a filename where to save the incomming data. Key in the name (e.g. flgtdata.b) and then press *Enter.* Type now *'G1500' enter* and the downloading will start
Be careful: if you make a typing mistake while keying the G1500, type *backspace* once and after the prompt reappairs, try again. Dont't press the enter key, you can loose the data.

8. Closing the file: when the data has been completely transfered, the help screen of the plane computer, will appeair again. press the *Escape* key in order to close the data file.

9. Exiting the commmunication program: Exit Procomm with *ALT-X*.

10. Checking the integrity of the file: Check that the downloaded data was correctly saved into the file, if not you can repeat the downloading operation. If the data was succesfulyy saved, you may switch the plane computer off and make a copy of the downloaded data file to a diskette (e.g. copy flgtdata.b a:)

11. Translating the binary data to decimal: In order to get a Tab separated decimal file to be used in a spreadsheet you have to use the Bin2Dec program. Type *Bin2Dec <filename.b> <filename.d> [-H]* where the first filename is the file with your recently downloaded data, the second one, is the file where the decimal data will be saved and the optional -H will tell the program to translate half the data in order to get a file usable in certain spreadsheets that don.t allow large sheets. After the Bin2Dec has finished (about 2 and half minutes for a 60K binary file using a 28ms Disk) you should also save a copy of the decimal data to a floppy disk. (e.g. copy flgtdata.d a:)

12 .The downloading procedure has finished: At this point, you have a file which can be used with almost every spreadsheet in order to process and analize the results of the filght.

Bin2Dec: Technical specifications

The purpose of the **Bin2Dec** program is to translate a file which contains the downloaded data of an experimental flight in binary format, to a tabs separated file which contains the same data in decimal format.
The computer at the plane registers in its RAM, 16 values 20 times per second. Every value may range from 0 to 255 and because of that, one byte is recquired for every one of them.
After the plane has landed the data at the memory is virtualy dumped into the computer and for the data to be usable in a spreadsheet, every byte of the downloaded data should be translated to the decimal number it represents. (e.g 10001001->137)
The **Bin2Dec** program reads every byte of the downloaded data and using a lookup table of obviously 256 strings, it translates it to the appropriate one. Every 16 bytes, the program writes a CRLF to the decimal file for the spreadsheet to start a new line. And in this way, we get a file with about 3600 lines of 16 tab separated decimal numbers which is ready to be used in almost any spreadsheet program (e.g. Excel, Quatro)
Before the implementation of the **Bin2Dec** program, the translation was performed at the plane and the data was transmitted in decimal format, but this caused the data tarnsfer to be a lengthy operation. Now,with this new technique, the figures are as follows:
Data acquired: 16 bytes x 20 times per sec x 180 sec = 60 Kbytes
Downloading:
At 9600 bauds = 960 bytes/sec = 60 Kb/min -> Data transfer 1 Min
At 4800 bauds = 480 bytes/sec = 30 Kb/min -> Data transfer 2 min.
The **Bin2Dec** program takes about 2 minutes for the translation of a 60 K file using a 386 computer with a disk of 28ms
This makes a total of about 3 minutes for getting the usable data

It's recommended to save the spreadsheet document once the data is dumped into it, in order to get a file in the custom format of every spreadsheet and in this way reduce the time recquired to open the file every time you want to process the results.

Appendix C: Basic Thrust-Vectoring Propulsion System/Control Weights

Fan + Fan Holder + Piston engine, Tune Pipe;	1200 gr
2 systems:	2400 gr
11 Servos; 60 x 11	660 gr
R/C Rec.	100 gr
Ni/Cad Bat. for onboard Computer	180 gr
Gyros and Radio Bat.: 190 x 2	380 gr
Fixed Landing gear	1000 gr
Electrical wires	150 gr
Mechanical "rods"	100 gr
Onboard computer	150 gr
3 Gyros (130 x 3)	400 gr
Fuel	600 gr
Fuel Vessels	120 gr
Velocity probe electrical generator	30 gr
2 Potentiometers	60 gr
	6320 gr

For Uninstrumented Model (less)

- 3 servos	180 gr
- Compt. Bat	180
- Gyros Bat.	90
- Computer	150
- 3 Gyros + probes	400 + 60
	1160 gr

+ structure = 14.4 Kg. (31.6 lb.)

Appendix D :

Terminology

Snap TV-roll reversals/stops and RaNPAS/reversals/stops by PST, or by PSM, may cause some **physiological effects** on the pilot. The physiological effects of various g loads are:

1) Difficulty of motion of body and limbs because of the weight increase;
2) Circulatory dysfunction concomitant with blood pooling, resulting in blackout and tissue hypoxia on the one hand and congestion on the other;
3) Displacement of viscera and other moveable parts;
4) Structural damage.

Tolerance generally means "**time until loss of consciousness at a given g load**". Tolerance to a given g value depends upon the duration and the direction of acceleration with respect to the body. When acceleration is from feet to head it is called "**positive**", when from head to feet it is "**negative**"; when from front to back or back to front it is **transverse**.

Human tolerances to acceleration at various rates show that for **0.1 – 0.3 sec** duration of a "**g-onset**", the typical times for the onset of TV-agility, the tolerances are:

3.0 – 7.5g for negative g loads.

7.0 – 10g for positive g loads.

Rotation, whether about one's own axis or some other, produces "motion sickness", especially when the subject must, in addition, move his head in some manner other than straight up and down. PST-TV-roll reversal/stops and TV-RaNPAS-rotations generate rapid rate-of-change in **sidewise g loads,** in positive and negative g loads, and in **centrifugal** g

loads. Thus, during TV-RaNPAS, it is mainly the rapid initiation of rotation and its quick stops that generate sidewise, and positive and negative g loads on the pilot. **For negative PST-RaNPAS the limit is 3.0 - 7.5 g.**

A relevant factor to 2nd-derivatives of the velocity vector during rapid TV-roll reversals/stops is as follows. In each ear there are 3 fluid-filled semicircular canals which are set in three planes **at right angles** to one another. In the absence of visual cues, the brain interprets stimuli arising from the semicircular canals in the following manner -

a - Constant velocity as "rest".

b - Acceleration as "movement".

c - Time-rate-of-changes in acceleration as "acceleration".

Furthermore, during straight and level longitudinal acceleration the pilot feels a false sensation of "**pitch-up**" change in attitude. During straight and level deceleration the pilot feels a false "**pitch-down**" change in attitude. Moreover, sudden linear acceleration - catapult, snatch, or "rocket launch" - produces the sensation of "**rotating backward**", heels over head, while sudden linear deceleration - crash impact, or arrester wires, produces the sensation of "**rotating forwards**", head over heels. Nodding movements of the head occuring whilst other rotational movements are taking place in a different plane, can give rise to considerable mental confusion and lead to disorientation.

APPENDIX E

Estimation of 1/7-Scale F-15 Model G_z-load During "Cobra" Maneuver

by Dr. V. Sherbaum

Estimation of maximum possible G-load during "cobra" maneuver execution is calculated. It is shown that the maximal G_z-load values achieved during F-15 1/7-scale aircraft flight tests will not be more than 7.2 G's.

Approximate PST-TV F-15 Model Data

M = Mass = 15 kg,

I_{yy} = Moment of inertia = 5.02 kg m^2,

T = Thrust = 80 N,

δ_v = TV-vane pitch angle = 30 deg.

CG = Center of gravity point,

P = Pilot position point (about 1 m ahead of from center of gravity),

B = Thrust force application point (about 1 m after CG point).

The following estimations are for a zero static stability margin.

It is presumed that the center of gravity motion is rectilinear one. Where the acceleration of one point (center of gravity point, for example) is known, the acceleration

of any other point (cf. P in Figs. 1 and 2) may be evaluated from

$$\bar{W}_P = \bar{W}_{CG} + \bar{W}_{PCG}^{\,n} + \bar{W}_{PCG}^{\,t}, \tag{1}$$

where

$\bar{W}_{CG}$ = center of gravity acceleration,

$\bar{W}_{PCG}^{\,n}$ = normal acceleration of point P (pilot's place) relatively to center of gravity,

$W_{PCG}^{\,n} = q^2 * PCG$, where q is the angular velocity,

$W_{PCG}^{\,t}$ = tangential acceleration of point P relatively to CG.

$W_{PCG}^{\,t} = \dot{q} * AO$, $\dot{q}$ = angular acceleration.

The causes of the center of gravity acceleration are a drag force and a thrust-vectoring force. One can see (cf. Fig.2) that for P point tangential acceleration and acceleration through the drag force have the same direction during "pitch-up" maneuver, therefore acceleration to the z-axis direction for P point is

$$W_P^{\,z} = W_{CG} + W_{PCG} \tag{2}$$

Calculation of the G - load

When our model velocity is taken as 35 m/sec, the wings drag force may be estimated as a flat plate's drag at AoA = 90 deg., (cf. Fig.2), vis.,

$$F_{drag} = C_D \rho S V^2/2 \quad , \qquad (3)$$

where

S = wings area = 0.74 m^2,

ρ = air density = 1.2 kg/m^3 ,

V = aircraft velocity = 35 m/s,

C_D = drag coefficient for a flat plate that is normal to flow direction = 1.8 [1].

Thus, from expression (3) drag force is

$$F_{drag} = 1.80 \; 0.74 \; 1.2 \; 35^2/2 = \underline{979 \text{ N}} \qquad (4)$$

Prior to the "cobra" reversal point the Z-component of T force has the opposite direction relatively to drag force one and it value is

T_z = T*sin 30= 80 0.5 = 40 N = T_V

Hence, the acceleration of the center of gravity is

$$W_{CG}^{\,z} = (F_{drag} - T^z) \,/\, M =$$
$$(979 - 40) \,/\, 15 = 62.6 \; m/s^2 \qquad (5)$$

Hence, $W_{PCG}^{\,t}$ acceleration may be calculated from

$$W_{PCG}^{t} = \dot{q} * PCG \qquad (6)$$

We next assume that the δ_v-reversal is conducted instantaneously thereby providing maximum moment M_{CG}^{max},

$$\dot{q} = M_{CG}^{max} / I_{yy} \qquad (7),$$

where torque is

$$M_{CG} = d_{BCG} * \mathrm{Sin}30 * T = 1 * 0.5 * 80 = 40 \text{ Nm.}$$

(It is assumed that the drag force does not generate torque relatively to the center of gravity).

Hence, from (7)

$\dot{q} = 40 / 5.02 = 8.0$ rad/s^2= 457 deg/sec^2

and from (6)

$W_{PCG}^{t} = 7.97*1 = 7.97$ m/s^2.

The total acceleration in the Z-axis direction is therefore (expression 2)

$-W_P^z = 62.6 + 8.0 = 70.6$ m/s^2 or 7.2 g. !

Consequently, 7.2 G is the maximal G_z-load value extractable from the "cobra" maneuver with a F-15 thrust-vectoring 1/7-scale model.

Dynamic scale factors (DSF) must be used to

find corresponding velocity for full-scale F-15 [2]. It is for velocity:

$$V_{aircraft} = V_{model}\ L^{0.5}$$

where L is the linear-scale-factor, L = 7.

So, for F-15

$$V_{aircraft} = V_{model}\ L^{0.5} = 35\ 7^{0.5} = 92.6\ m/s,$$

or Mach number = 0.28.

For full-scale F-15 Mach number = 0.4, G-load = 13.8.

Real G-load will be less than these data and more accurate results maybe calculated from mathematical descriptions [2].

Note: For non-zero static stability margin, when the center of pressure is forward of the center of gravity, the pitch-up torque and pitch-up angular rates increase during the PST maneuver. The opposite effect is expected during pitch-down maneuver.

REFERENCES

1. Paul K. Chang.
Separation on Flow , Pergamon Press, 1970.
2. Tailless Vectored Fighters. Report to USAF.
Jet Propulsion Laboratory, TIIT, 1991.

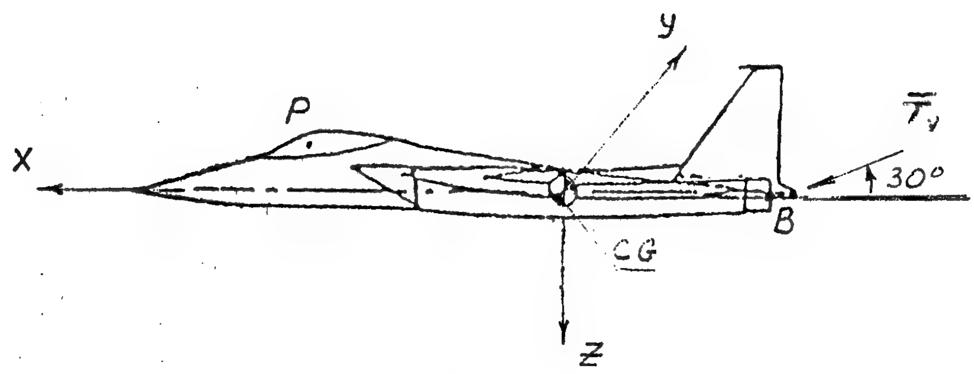

y
P
X
$\bar{T}_v$
30°
B
CG
Z

Fig. 1

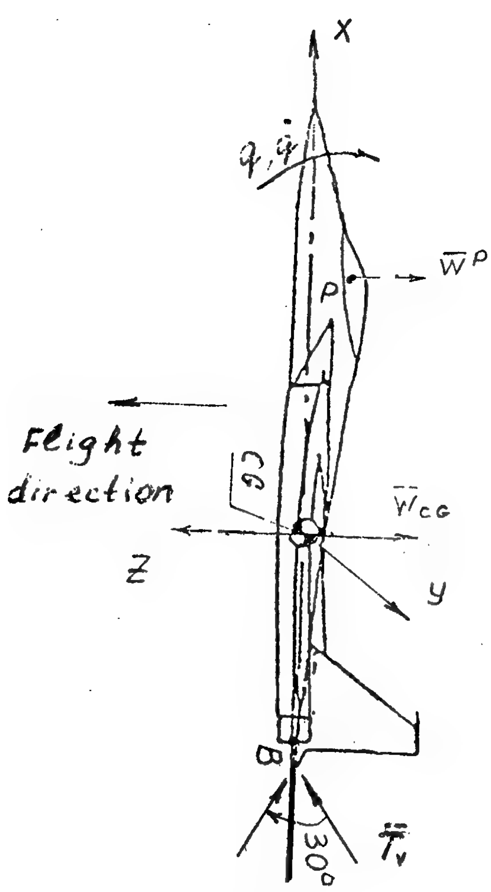

X
q, q̇
$\bar{W}^P$
P
Flight direction
CG
$\bar{W}_{CG}$
Z
y
B
30°
$\bar{T}_v$

Fig. 2

APPENDIX F: PILOT'S G-LOAD SIMULATION/MEASUREMENTS

BY DR. MICHAEL LIKHTSINDER

ABSTRACT. A G-load measurement system is described. It uses accelerometer readings for both pilot acceleration and the acceleration due to gravity. Since vehicle engines generate noise a noise filter optimization was executed. A two-elements RC-filter calculation is described. The filter decreases engines noise levels by a factor of 100.

Measurement channels sensitivities determination, measurement channels calibration and flight simulation methodology are described. Simulation results are shown with and without filters, with one or two engines operating and with nonoperating engines.

1. DEFINITIONS

G-LOAD is a vector, viz.,

$$\vec{G} = \frac{\vec{F}_r}{m g}, \qquad (1)$$

where $\vec{F}_r$ =the pilot support (arm-chair, floor, belts) force reaction,

m =pilot mass.

Support force reaction is applied to the

pilot. Pilot flight weight is force vector ~~applied~~ to the support. It equals the support force reaction module and has the opposite direction to the support force reaction.

The G-load components in Body Axes are:

$$G_x = \frac{F_{rx}}{mg}, \qquad (2)$$

$$G_y = \frac{F_{ry}}{mg}, \qquad (3)$$

$$G_z = \frac{F_{rz}}{mg}, \qquad (4)$$

where F_{rx}, F_{ry}, F_{rz} are the $\vec{F}_r$ components in body axes.

G-load components coincide with axis positive direction (see p. 9), when

G_x =blood flow from back to chest,

G_y =blood flow from left side to right side,

G_z =blood flow from feet to head.

It is desired to know SUPPORT FORCE REACTION for PILOT'S G-LOAD determination.

The pilot's motion equations in Body Axes are :

$$m\vec{a}=\vec{F}_r+m\vec{g}\ , \tag{5}$$

$$ma_x=F_{rx}+mg_x, \tag{6}$$

$$ma_y=F_{ry}+mg_y, \tag{7}$$

$$ma_z=F_{rz}+mg_z, \tag{8}$$

where $\vec{a}$=acceleration, a_x, a_y, a_z =acceleration components.

From (6)-(8):

$$\begin{aligned} F_{rx}&=m(a_x-g_x),\\ F_{ry}&=m(a_y-g_y),\\ F_{rz}&=m(a_z-g_z). \end{aligned} \tag{9}$$

From (2)-(4) and (9):

$$G_x=\frac{a_x-g_x}{g}, \tag{10}$$

$$G_y=\frac{a_y-g_y}{g}, \tag{11}$$

$$G_z=\frac{a_z-g_z}{g}. \tag{12}$$

On the ground in a static position

$$G_x=0,\ G_y=0,\ G_z=-1.$$

2. G-LOAD MEASUREMENTS.

2.1. What does an aircraft accelerometer measure?

Standard accelerometer is a single degree of freedom oscillating system (Fig.1, Fig.2).

Three accelerometers are installed in a flying vehicle so that the sensitivity axes are directed along the Body Axes Oxyz. Accelerometer seismic mass motion along the x-axis equation is

$$m\ddot{x}+p\Delta\dot{\ell}+K\Delta\ell=mg_x, \qquad (13)$$

where

m=accelerometer seismic mass,

p=damper coefficient,

k=elastic coefficient of a cantilever beam,

$\Delta\ell$=seismic mass deflection along the x-axis.

Expressing accelerometer output voltage as

$$u_x=C\cdot\Delta\ell=C\cdot(mg_x-m\ddot{x}-p\cdot\Delta\dot{\ell}), \qquad (14)$$

where $|p \cdot \Delta l| \ll |m g_x - m\ddot{x}|$, hence (14) may be rewritten as

$$u_x = C_{ac}^{x} \cdot (a_x - g_x). \tag{15}$$

$C_{ac}^{x} = C \cdot m$ is the accelerometer sensitivity to acceleration $[V/(m/s^2)]$.

By analogy to (15) one may write for the other accelerometers:

$$u_y = C_{ac}^{y} \cdot (a_y - g_y), \tag{16}$$

$$u_z = C_{ac}^{z} \cdot (a_z - g_z). \tag{17}$$

By comparing (15)-(17) with (10)-(12) one may conclude: EACH ACCELEROMETER READING CORRESPONDS EXACTLY TO THE G-LOAD COMPONENTS IN THE BODY AXES SYSTEM.

2.2. G-load measurement with noise filtration

The aircraft engines generate noise which affects G-load measurements. An engine noise diagram is depicted in Fig.XII. Noise maximum levels reach ± 4.8. It is therefore

necessary to use a filter.

Filter calculations

Engine rotation speed is 20,000 RPM, i.e.,

$$f_{en} = \frac{20,000}{60} = 333\,Hz \quad or \quad \omega_{en} = 2091\,\frac{rad}{sec}.$$

Suppose there is linear acceleration exponential transient approximation during a pitch maneuver, viz.,

$$a_z = a_{mz}\left(1 - e^{-\frac{t}{\tau_z}}\right),$$

where τ_z =aircraft time constant during the pitch maneuver.

Its minimum transient time, $t_{min} = 0.75\,sec$, i.e.,

$$\tau_z \approx \frac{t_{min}}{3} = 0.25\,sec.$$

It is necessary for the filter a time constant $\tau_f = \frac{\tau_z}{5} = 0.05\,sec$ to minimize process distortion.

The filter schemes are shown in Fig.3.

Accelerometers output impedance $R_{out} \approx 500\,\Omega$ therefore

$$\tau_f = R_\Sigma C_\Sigma = (500+4700)\cdot 2\cdot 4.8\cdot 10^{-6} = 0.05\,sec.$$

Writing the filter transfer function as

$$W(s)=\frac{1}{\tau_f s+1}, \qquad (18)$$

where s=Laplas complex argument, the filter frequency response becomes

$$K(\omega)=\left|W(j\omega)\right|=\frac{1}{\sqrt{1+\omega\tau_f}}. \qquad (19)$$

From (19) one obtains $K(\omega)$ =0.01 if ω =2091rad/sec and τ_f =0.05sec.

Hence, the engine noise level is decreased by a factor 100.

3. G-LOAD MEASUREMENT CHANNELS CALIBRATION

3.1. Model EGA-125-10D accelerometers initial characteristics are shown in table 1

Accelerometers initial sensitivity corresponds to input voltage U=15v. The vehicle battery nominal voltage U_{bat} =4.8v, so that the accelerometer sensitivity is calculated by

$$C_{ac}=C^{i}_{ac}\frac{U_{bat}}{U}=C^{i}_{ac}\frac{CN_{bat}}{U\cdot C^{bat}_{ms}}=$$

$$=0.0015\,C^{i}_{ac}\cdot CN_{bat}, [mV/g], \qquad (20)$$

where

C^{i}_{ac} =initial accelerometer sensitivity, [mv/g],

C_{ac} =accelerometer sensitivity with battery nominal voltage 4.8v,[mv/g],

C^{bat}_{ms} =44.05 =measurement system sensitivity to battery voltage,[CN/v],CN=computer number,

CN_{bat} =battery voltage channel computer number (channel 0F).

The measurement system sensitivity to acceleration is therefore

$$C^{a}_{ms} = C_{ac} \cdot C^{v_a}_{c} =$$
$$= 0.019 \cdot C^{i}_{ac} \cdot CN_{bat}, \ [CN/g]. \quad (21)$$

In (21) $C^{v_a}_{c} = \frac{255}{20} = 12.75$ [CN/mv] is sensitivity of the computer channel measuring accelerometers output voltage.

The G-load is now computed from

$$G_i = \frac{CN_{ac}}{C^{a}_{ms}} =$$
$$= 52.63 \frac{CN_{ac}}{C^{i}_{ac} \cdot CN_{bat}}, \text{[dimensionless]}, \quad (22)$$

where CN_{ac} is the corresponding acceleration

channel computer number.

Initial accelerometer sensitivity, C^{i}_{ac} ,for each accelerometer is provided in table 1.

Calculated by (21) each measurement channel sensitivity is shown, for U_{Bat}=4.8v, in table 1.

3.2. <u>G-LOAD measurement channel</u>

calibration may be executed with higher accuracy by the following manner.

Accelerometer output voltage is twice measured in a static state. The first time accelerometer sensitivity axis is directed along vertical "up" (G=+1), while in the second the same accelerometer axis is directed along the vertical "down" (G=-1).

G -load signs are determinated by (10)-(12).

Then the measurement system sensitivity corresponding battery voltage U_{Bat}=4.8v

$$C^{G_i}_{ms} = \frac{CN^{+}_{aci} + CN^{-}_{aci}}{2} \cdot \frac{211 \cdot 2}{CN^{+}_{Bat} + CN^{-}_{Bat}} =$$

$$= \frac{CN^{+}_{aci} + CN^{-}_{aci}}{CN^{+}_{Bat} + CN^{-}_{Bat}} \cdot 211 , \qquad [CN/g]. (23)$$

The measurement system sensitivity corresponding to other battery voltages are

$$C_{ms}^{G_i} = C_{ms}^{G_i}\Big|_{U_{Bat}=4.8V} \times \frac{CN_{Bat}}{211} =$$

$$= \frac{CN_{aci}^{+} + CN_{ac}^{-}}{CN_{Bat}^{+} + CN_{Bat}^{-}} \cdot CN_{Bat}, [CN/g]; (24)$$

G-load components are computed by

$$G_i = \frac{CN_{ac}^{i}}{C_{ms}^{G_i}} =$$

$$= \frac{CN_{Bat}^{+} + CN_{Bat}^{-}}{CN_{aci}^{+} + CN_{aci}} \cdot \frac{CN_{ac}^{i}}{CN_{Bat}}, (25)$$

Hexadecimal notation computer numbers are written for six aircraft static positions corresponding to six Body Axes System positions shown in table 2. G-load ideal volumes ±1 for each case are shown in table 2.

The measurement system sensitivity corresponding to U_{Bat} =4.8v is calculated by (23) for hexadecimal notation volumes in table 2. The calibration results are shown in table 1 and in Fig.IX-Fig.XI. When shifts of G are about t=0, then in Fig.IX-Fig.XI

$$X = \frac{Comp.No(G_x) - Comp.No(G_x)\big|_{t=0}}{Comp.No(V_{Bat})},$$

$$X = \frac{Comp.No(G_y) - Comp.No(G_y)\big|_{t=0}}{Comp.No(V_{Bat})},$$

$$X = \frac{Comp.No(G_z) - (\pm Comp.No(G_z)\big|_{t=0} + 28.6}{Comp.No(V_{Bat})}$$

4. ACCELERATION MEASUREMENT CHANNELS CHECKING (FLIGHT SIMULATION)

X-axes acceleration measurement channel is checked by the following manner. Three lines A,B,C are drawn on the floor (see Fig.4). The aircraft model is oscillated in the X direction with amplitude of 0.5m and a frequency of 1Hz, viz.,

$$x = 0.5 \cdot \sin 2\pi t, \quad [m], \qquad (26)$$

employing linear acceleration

$$a_x = \ddot{x} = -0.5(2\pi)^2 \sin 2\pi t =$$

$$= 19.7 \sin 2\pi t, [m/sec^2]. \qquad (27)$$

Acceleration amplitude is measured in the X-acceleration channel and compared with computer number CN_a^x =82 (in hexadecimal notation CN_a^x =52), when U=4.8v.

Y,Z- acceleration measuring channels are similarly simulated. In these cases the following oscillations are used:

$$y = 0.5 \sin 2\pi t, [m],$$

$$a_y = 19.7 \sin 2\pi t \; [m/sec^2], \qquad (28)$$

$$z = 0.5 \sin 2\pi t, [m]$$

$$a_z = 19.7 \sin 2\pi t, [m/sec^2]. \quad (29)$$

The simulation results are shown in Fig. III–Fig. VIII.

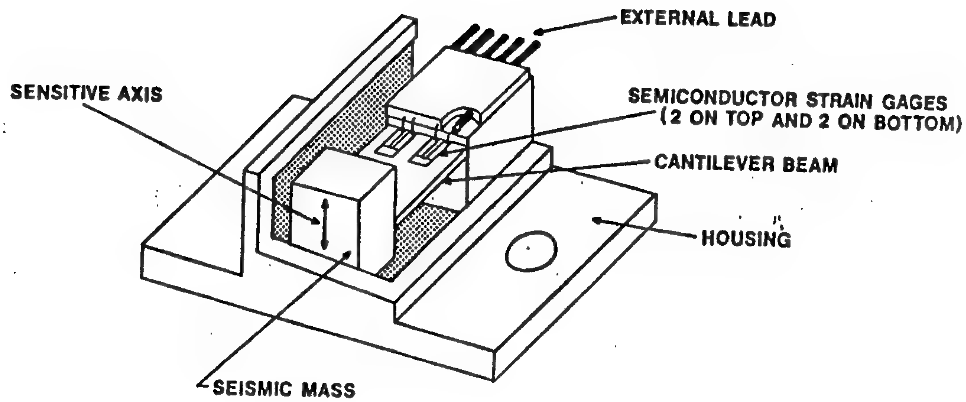
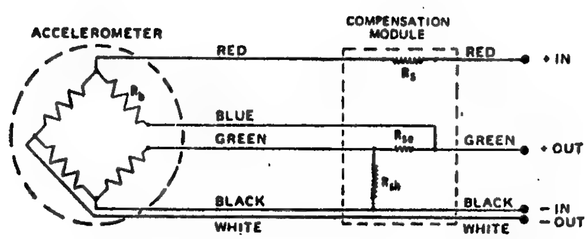


Fig. 1

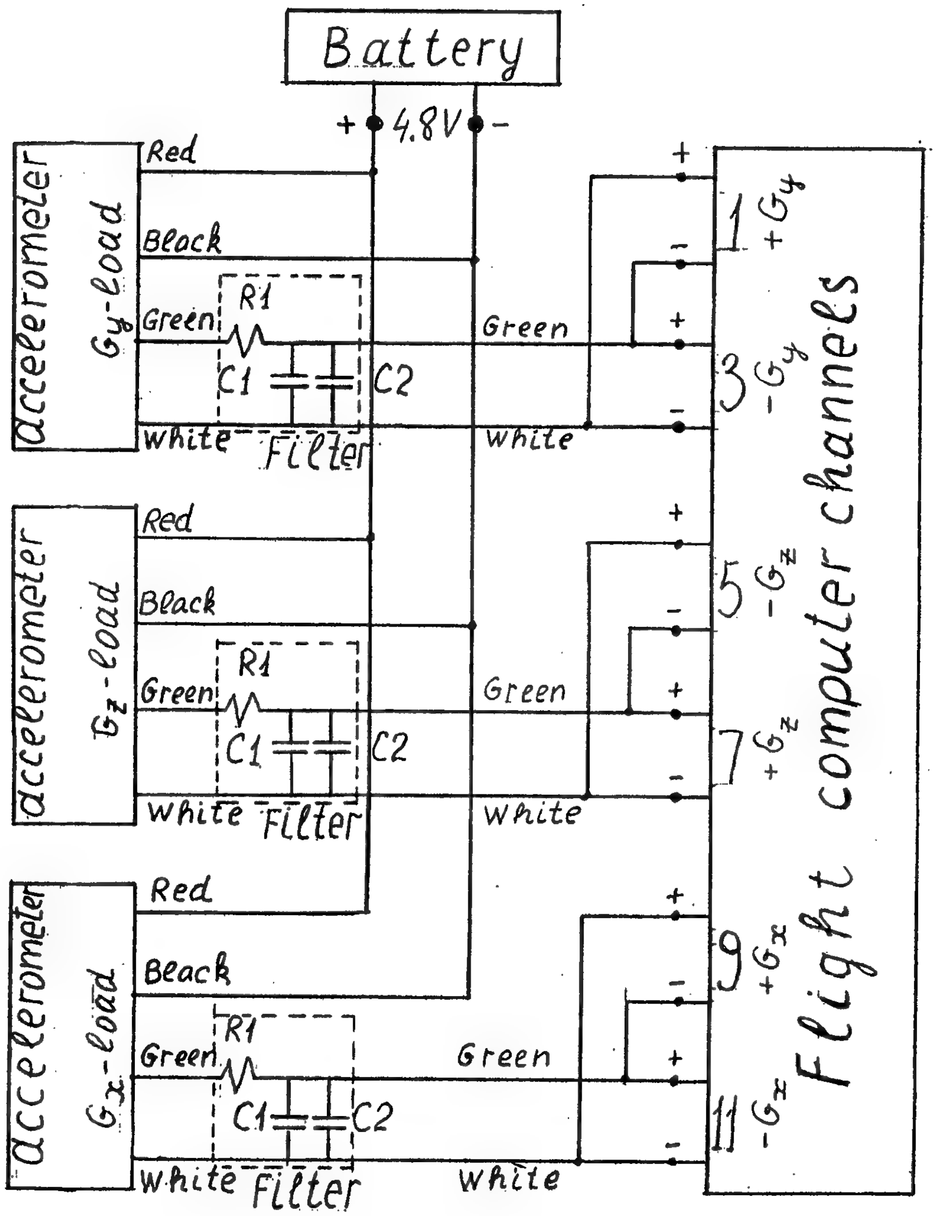


R_b : Semiconductor Strain Gage
R_s : Sensitivity Compensation Resistor

R_{se} : Zero Offset Trim Resistor

R_{sh} : Zero Shift Compensation Resistor (sometimes located between Red and Blue or Black and White leads - usually between 10,000 and 100,000 ohms)

Fig. 2



R1=4.7KΩ, C1=C2=4.8μF.

Fig.3: G-load measurement channels.

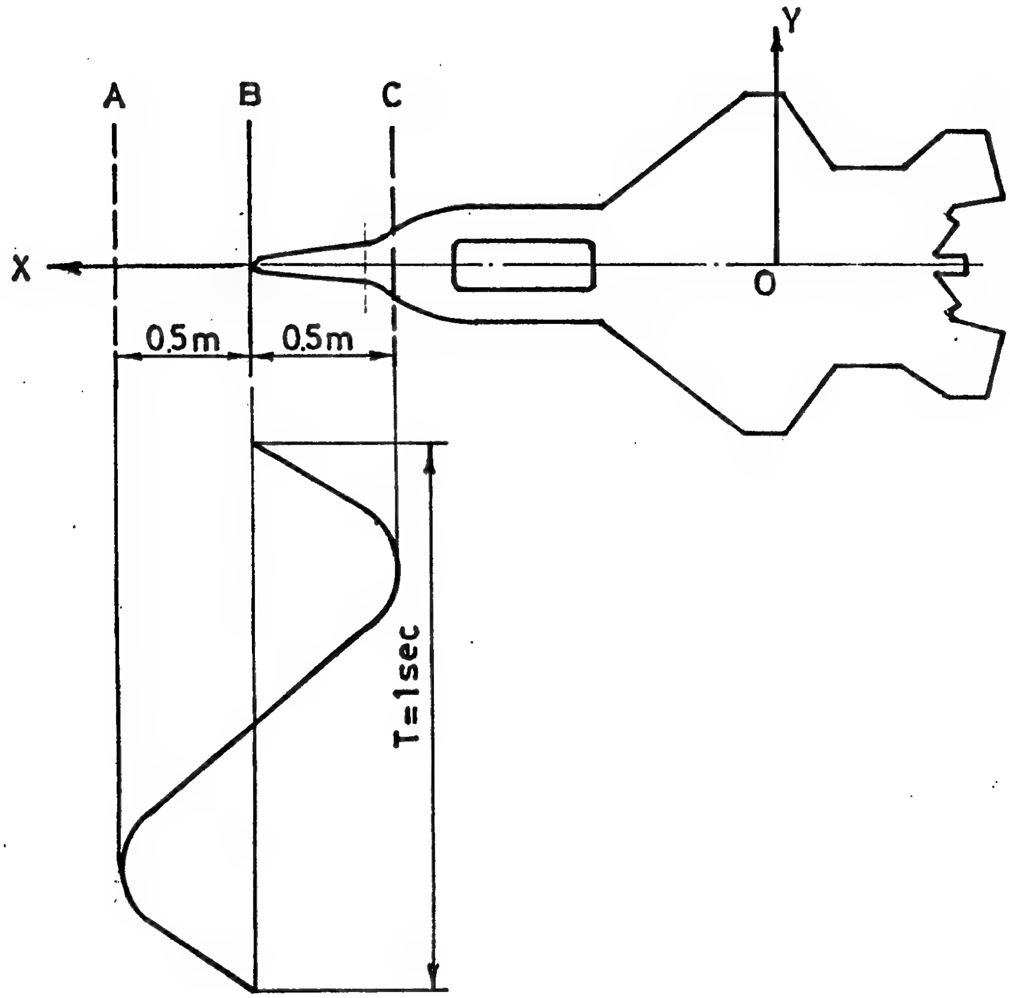
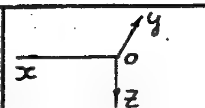
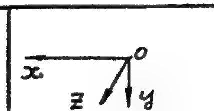
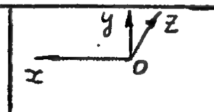
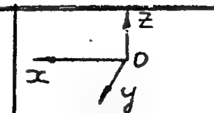
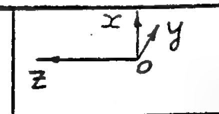
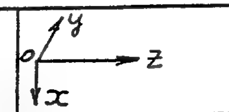


Fig. 4

Table 1

Axis	Initial range	Initial input voltage	Initial sensitivity	Impedance Ω		G-load measurement system				
						Calculation			Calibration	
				input	output	Sensitivity		Range	Sensit	Range
	g	VDC	mV/g			mV/g	CN/g	dim-less	CN/g	dim-less
X	± 10	15.0	9.75	811	480	3.16	39.0	±6.5	34.6	± 7.4
Y	± 10	15.0	12.30	858	463	3.90	49.4	±5.2	46.9	± 5.4
Z	± 10	15.0	9.07	801	512	2.87	36.2	±7.0	28.1	± 9.1

Table 2

G-load sensitivity axis									
x axis	Ideal G-load			0	0	0	0	+1	−1
x axis	Channel	$+G_x$	09	0;0;0;0;0	0;0;0;0;0;	0;0;0;0;0	0;0;0;0;0	0F;0F;0E;0F;0F	0;0;0;0;0
x axis	Channel	$-G_x$	0B	0;0;0;0;0	0A;0A;09;09;09	0;0;0;0;0	04;03;03;02;03	0;0;0;0;0	35;38;37;39;39
y axis	Ideal G-load			0	−1	+1	0	0	0
y axis	Channel	$+G_y$	01	10;10;10;10;10.		47;46;46;47;46	0C;0B;0B;0B	01;01;03;02;01	0F;10;0F;0F;11
y axis	Channel	$+G_y$	03	0;0;0;0;0.	1A;1A;18;19;1A	0;0;0;0;0.	0;0;0;0;0.	0;0;0;0;0	0;0;0;0;0
z axis	Ideal G-load			−1	0	0	+1	0	0
z axis	Channel	$-G_z$	05	11;14;11;12;12	0;0;0;0;0	0;0;0;0;0	0;0;0;0;0	0;0;0;0;0	0;0;0;0;0
z axis	Channel	$+G_z$	07	0;0;0;0;0	0;0;0;0;0	0C;0B;09;0B;0C	28;2A;29;28;27	0;0;0;0;0	06;06;06;06;05
Battery voltage	Channel		0F	DA;D9;D3;D6;D3	DC;D8;D9;D9;D6	D5;D7;D8;D9;D8	D7;DA;D7;D9;DC	D8;D4;D5;D6;D2	D6;D4;DA;D4;D5

APPENDIX G: FLIGHT BATTERIES OPTIMIZATION

By Dr. Michael Likhtsinder

1. Introduction

One may divide aircraft electrical circuits into three groups shown in table 1. There are two types of DC batteries shown in table 2. One may use battery of type 2 as two batteries with voltage 4.8v. In this case each battery capacity is 600mAh.

It is desired to optimize the flight batteries about weight, time and cost.

2. Batteries optimization

Three schemes of the batteries and the control system, the measurement devices and the computer combinations are shown in table 3. The lowest summary batteries weight 190gr corresponds to scheme 7 when maximum time is 58min and reliability is middle.

Schemes 3 reliabilities may be raised with help of a safety-lock in the computer circuit. In this case control system's battery is protected and flight control ability is remained .

Remarks:

1. Old batteries capacities are decreased hence maximum time in table 3 are decreased.

2. It is necessary to charge

-battery type 1 (4.8v) due to 30 hours,

-battery type 2 (9.6v) due to 15 hours,

when source current is 40mA.

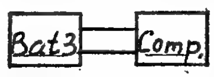
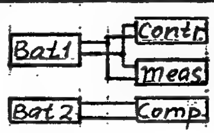
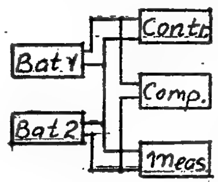
Table 1

Group	Electric device	Quality	Voltage DCV	Current, mA One	Current, mA All	Current, mA Group
Measurement	Gyro	3	4.8	100	300	317
	Acceler.	3	4.8	5	15	
	α-probe	1	4.8	1	1	
	β-probe	1	4.8	1	1	
Control	Receiver	1	4.8	50	50	70
	Servo	10	4.8	2 mid value	20	
Comp.	Computer	1	9.6	300	300	300

Table 2

Battery type	Voltage DCV	Capacity mAh	Weight gram	Cost $
Type 1	4.8	1200	180	50
Type 2	9.6	600	190	40

Table 3

Scheme	No	Type of battery	One battery: Voltage V	Current mA	Time min	Weight gr.	Cost $	General: Weight gr.	Time min	Cost $	Reliability
	1	Bat1=Type1	4.8	70	1028	180	50	550	120	130	high
		Bat2=Type1	4.8	317	227	180	40				
		Bat3=Type2	9.6	300	120	190	40				
	2	Bat1=Type2	4.8	70	514	95	20	380	113	80	high
		Bat2=Type2	4.8	317	113	95	20				
		Bat3=Type2	9.6	300	120	190	40				
	3	Bat1=Type2	4.8	70	514	95	20	455	120	110	high
		Bat2=Type1	4.8	317	227	180	50				
		Bat3=Type2	9.6	300	120	180	40				
	4	Bat1=Type1	4.8	387	186	180	50	370	120	90	low
		Bat2=Type2	9.6	300	120	190	40				
	5	Bat1=Type2	4.8	387	93	95	20	285	93	60	low
		Bat2=Type2	9.6	300	120	190	40				
	6	Bat1=Type1	4.8	370	194	180	50	360	116	100	middle
		Bat2=Type1	4.8	617	116	180	50				
	7	Bat1=Type2	4.8	370	97	95	40	190	58	80	middle
		Bat2=Type2	4.8	617	58	95	40				
	8	Bat1=Type2	4.8	370	194	95	20	275	117	70	middle
		Bat2=Type1	4.8	617	117	180	50				

Fundamental Concepts of Vectored Aircraft

Benjamin Gal-Or*, Jet Propulsion Laboratory, Faculty of Aerospace,

Technion - Israel Institute of Technology, Haifa, 32000, Israel, and

Daniel D. Baumann**, Flight Dynamics Directorate, Wright Laboratories,

WPAFB, Dayton, Ohio, USA.

* Professor and Head of Laboratory, Past Member of AIAA Airbreathing Commt.

** USAF Capt. and Flight Test Manager of the thrust-vectored F-15 STOL/Maneuverability Technology Demonstrator Program. USAF/WOE Visitor to JPL/Technion.

Fundamental concepts of post-stall [PST], pure or mixed thrust-vectored-controlled [TVC] aircraft are defined and employed as 'ideals' to maximize TVC-rates during PST, supermaneuvers. Employing these concepts, a proposed unified mathematical phenomenology defines yaw-pitch, or roll-yaw-pitch TVC for maximized rates/moments and minimized critical delay-times during combat. Basic TVC rules and criteria are deduced and a new formulation/methodology for flying dynamically-scaled, TVC-prototypes, or PST-TVC-variants of F-15, F-16 and F-22 aircraft, is presented. Forbidden human space-time domains and critical time-derivatives that infer far-reaching consequenses for super-agile fighters are analyzed.

Nomenclature

AB = After burning,

AGA = Aircraft Gross Agility,

A-IFPC = Aircraft IFPC,

AoA = Angle of attack,

b = reference span, [m]

c = reference mean aerodynamic chord, [m],

C_D = drag coefficient, dimensionless,

CG = center of gravity, % mean aerodynamic chord,

C_{fg} = engine nozzle thrust coefficient. Dimensionless, [cf. eqs. 16-18],

C_L = lift coefficient, dimensionless,

C_l = rolling moment coefficient, dimensionless,

$C_{l\beta}$ = rolling moment derivative with respect to sidslip angle, 1/rad,

$C_{l\delta a}$ = aileron effectiveness derivative, 1/rad,

$C_{l\delta e}$ = stabilator effectiveness derivative, 1/rad,

$C_{l\delta\Delta e}$ = differential stabilator effectiveness derivative, 1/rad,

$C_{l\delta r}$ = rudder effectiveness derivative [variation of rolling moment coefficient with respect to rudder angle], 1/rad,

C_{lp} = roll damping derivative, 1/rad,

C_{lr} = rolling moment derivative with respect to yaw rate, 1/rad,

C_m = pitching moment coefficient, dimensionless

C_{mo} = basic pitching moment coefficient, dimensionless

C_{mq} = pitching moment derivative with respect to pitch rate, 1/rad,

C_n = yawing moment coefficient, dimensionless

$C_{n\beta}$ = yawing moment derivative with respect to sideslip angle, 1/rad,

$C_{n\beta*}$ = yawing moment derivative high angle-of-attack increment with respect to sideslip angle, 1/rad,

$C_{n\delta a}$ = yawing moment derivative with respect to aileron deflection, 1/rad,

$C_{n\delta e}$ = yawing moment derivative with respect to stabilator deflection, 1/rad,

$C_{n\delta\Delta e}$ = yawing moment derivative with respect to differential stabilator deflection, 1/rad,

$C_{n\delta r}$ = rudder effectiveness derivative [variation of yawing moment coefficient with respect to rudder angle], 1/rad.

C_{np} = yawing moment derivative with respect to roll rate, 1/rad,

C_{nr} = yaw damping derivative, 1/rad,

C_{py} = side-center-of-pressure [for PSM in the y-direction],

C_x = longitudinal force coefficient, dimensionless,

C_y = side force coefficient, dimensionless,

$C_{y\beta}$ = side force derivative with respect to sideslip angle, 1/rad,

$C_{y\beta*}$ = asymmetric side force derivative high angle-of-attack increment with respect to sideslip angle, 1/rad,

$C_{y\delta a}$ = side force derivative with respect to aileron deflection, 1/rad,

$C_{y\delta e}$ = side force derivative with respect to stabilator deflection, 1/rad,

$C_{y\Delta e}$ = side force derivative with respect to differential stabilator deflection, 1/rad,

$C_{y\delta r}$ = side force derivative with respect to rudder deflection, 1/rad,

C_{yp} = side force derivative with respect to roll rate, 1/rad,

C_{yr} = side force derivative with respect to yaw rate, 1/rad,

C_z = normal force coefficient, dimensionless,

D = the distance from TV nozzle exit to aircraft C_{py}, [m],

D^* = the distance from TV nozzle exit to aircraft **CG**, [m],

D_{cpy} = the drag operating @ C_{py}, [kgf],

2D-CD = two-dimensional, converging-diverging nozzles,

g = gravitational constant, m/sec^2,

G_z, G_y, G_x = 'g-onsets' on the pilot in the respective body-axis coordinates, [m/sec^2],

IFPC = Integrated Flight Propulsion Control,

I_{ij} = inertia tensor (i,j = 1, 2, 3 or **x**, **y**, **z**); [kg-m^2],

I_X = moment of inertia about the roll axis, [kg-m^2],

I_{XY} = cross product of inertia between roll and pitch axes, [kg-m^2],

I_{XZ} = cross product of inertia between roll and yaw axes, [kg-m^2],

I_Y = moment of inertia about the pitch axis, [kg-m2],

I_Z = moment of inertia about the yaw axis, [kg-m^2],

L = linear scale factor defined by eqs. 20-22,

M = vehicle mass, [kg]; also Mach number, [dimensionless],

$\dot{M}$ = nozzle air mass flow rate, [kg/sec],

MGA = Model Gross Agility.

M-IFPC = Model Integrated Flight/Propulsion Control,

NPR = Nozzle pressure ratio, [dimensionless],

p = roll rate, [rad/sec],

PSM = pure sideslip maneuvers.

PST = post-stall,

PVA = Pure [thrust-] Vectored Aircraft,

q = pitch rate, [rad/sec],

$\bar{q}$ = dynamic pressure, employed with reference area **s**, $(1/2)\rho V^2$, [N/m^2],

r = yaw rate, [rad/sec]; or radius [m],

R = radius of gyration, [m],

s = reference aircraft surface area for dynamic pressure/force calculations, [m^2],

SACOM = Standard Agility Comparison Maneuver,

t = time [note that 'time' is 'compressed' by dynamic scale factors [Cf. eqs. 20 - 22], [sec],

T = actual [net] thrust, [cf. eqs. 16-18], [kgf]; also temperature, [deg. Kelvin],

T_i = ideal isentropic [net] thrust, [cf. eqs. 16-18] , [kgf],

$T_{x,y,z}$ = thrust-vectored components in the [body-axis] **x**-, **y**-, **z**- coordinates, [kgf],

T_V = pitch thrust vectoring component [equals T_z for **x**-aligned TV-engines], [kgf],

TV = thrust vectoring

TVC = thrust-vectoring control

V = true airspeed,[m/sec],

W = flying vehicle weight, [kgf],

Y = the distance from aircraft centerline to [split-type] TV-nozzle centerline, [m],

Greek

α = angle of attack, also AoA, deg, or rad,

β = angle of sideslip, deg, or rad,

δ_a = aileron surface deflection, [may be a differential angle], deg, or rad,

δ_e = elevator [stabilator] surface deflection, deg, or rad,

$\delta_{\Delta e}$ = differential elevator surface deflection, deg, or rad,

δ_r = rudder surface deflection, deg, or rad,

δ_{TV} = effective deflection angle of the jet during pitch and/or yaw thrust vectoring, [may be a differential angle during a TV-roll command], deg, or rad,

δ_v = effective pitch thrust-vectoring angle, [may be a differential angle during a TV- roll command], deg, or rad,

δ_y = effective yaw thrust vectoring angle, [may be a differential angle during a PSM-Yaw-command], deg, or rad,

ΔZ_{offset} = thrust offset, vanishes for x-aligned TV-engines, [m],

ϕ = bank angle, deg,

θ = pitch angle, deg,

ψ = heading angle, deg,

$\bar{\rho}$ = average density, [kg/m^3].

Introduction

Traditionally, jet engines have been considered to have little influence on flight-control theories, system designs, and effective combat agility. They were a-priori confined to provide brute, unvectored, forward force. The required moments for maneuverability and controllability were reserved for aerodynamic control surfaces, which are a-priori limited by external-flow regimes, and, hence, by the so-called stall barrier. This thinking has totally ignored the unprecedented control potentials of engine forces, especially beyond the so-called stall limit, i.e., during "impossible" Post-Stall [PST] maneuvers at extremely high pitch, yaw and roll rates. Consequently, in the past, aerodynamicists tended to develop theories in conjunction with only a rudimentary flight control role for engine thrust.

However, the increasing demands on PST-agility and supermaneuverability have recently begun a radical change in these attitudes. Almost suddenly it was realized that there is no

unified mathematical framework and criteria to define and quantify the new problems properly.

Therefore, new integrated methodology/phenomenology, basic criteria and PST-TV-yardsticks of effectiveness must be evolved, apparently from no verifiable base of low-risk, flight-tested data-bases. In turn, such attempts to revolutionize the mode of thinking of propulsion, aerodynamic, system design and flight-control engineers, may change a basic approach to aeronautical engineering education, theories and practice.

Reassessment of Aircraft/Missile Integration Concepts

The availability of PST vectored fighters, helmet-sight-aiming systems, all-aspect missile and the new generation of EW systems, requires reassessment of the optimal balance between aircraft agility and effectiveness, and the agility and effectiveness of missile/helmet-sight-aiming systems. Whatever is the aforementioned balance, high-performance fighter aircraft will gradually be based on improved thrust-vectored propulsion/maneuverability/controllability [1].

In offensive engagements it may mean the ability to point the nose/weapon at the enemy first during very short, computer/system inherent technology delay times. For instance, when both combatants use advanced, yaw-pitch, PST-TVC, it means the capability to exploit the inherent computer/system/missile-release delay-times, i.e., from pilot's decision-time to shoot till secure-locking/missile's-release-time, for simultaneous rapid bottom/nose-pointing toward/closer-to the target, so as to minimize missile's flight path/time to target.

This minimum-time rule increases kill-ratio probabilities to destroy the target prior its launching its weapon, for otherwise the probabilities of mutual destruction increase dramatically. Consequently, aircraft PST-agility must be well-integrated with missile's PST-agility and initial vectoring conditions. Such integrated PST-agilities, require reassessment of the combined aerodynamics of both aircraft and missile [1].

In defensive engagements it means dramatic increases of survivability, using, say, very-rapid, unconventional pure side-slips maneuvers without banking [see below]. Therefore, as a result of improved survivability and combat effectiveness, the fleet size, with respect to expected missions/threats, can be significantly reduced.

New Domains

Fig. 1 defines the new domains of pseudo-steady-state and transient TV-enhanced flight-control/envelopes. It schematically shows how TVC-power decreases with altitude, while increasing with speed up to a maximum value. Such potentials are represented by the **T/W** lines in the figure. Transitions from beyond visual range [**BVR**] to within visual range [**WVR**] engagements increase pilot's needs for transient TV-enhanced maneuverability in air-to-air and in air-to-ground tactics. TV-enhanced air-to-ground operations can involve new, pure, TV-controlled, sideslip maneuvers without banking/rolling to be stressed below. Hence, TV becomes a key combat element, especially in close-in combat engagements, in air-to-ground operations, in gaining STOL/safety qualities, in reducing optical, IR and radar signatures, and in reducing the fleet size. It also provides certain advantages under high subsonic and supersonic flight regimes, including lower drag and signatures [Cf. Ref. 1 and below]. The use of down-pitch TV may increase direct and supercirculation-generated lift, thereby slightly expanding the upper flight envelope, as depicted.

We therefore assert that in future aerial combat, once the engagement has closed to WVR, even under multitarget situations, PST-TVC-effectiveness becomes a critical capability to win and survive. Dramatic increases in performance and survivability in revolutionary air-to-ground tactics are also expected [1]. Hence, the fundamental concepts and efficiencies/limitations of PST-TV-aircraft must be well-formulated/flight-tested by means of dynamically-scaled and full-size prototypes.

Reassessment of Conventional Concepts

Thrust-vectored flight control [TVC] is either "pure" or "mixed". In pure TVC, the AoA-dependent moments generated by conventional control-surfaces, are entirely replaced by moments generated by rapidly-deflecting engine-exhaust jet(s), i.e., pure TV-aircraft can deliver top PST-control power/rates without recourse to ailerons, flaps, elevators, and rudders, and even the vertical tail-stabilizer may become redundant **[Figs. 3 and 4]**. Combined with the following methodology/phenomenology, these initial concepts can guide the

development of PST, super-agile, tailless vectored fighters.

Engine forces hardly change with external-aerodynamic flow regimes, especially when equipped with PST-vectorable inlets. Therefore, the control forces available for Pure Vectored Aircraft [PVA], remain highly effective even beyond the maximum-lift AoA **[Fig. 5]**. Therefore, PVA present the 'ideal' potential to maximize flight-control power and combat agility, even in the deep PST-domain. Hence, PVA concepts must be established as the highest standard, or as the 'ideal' reference to maximize flight-control-power and PST-controllability.

Such a standard must be based on verifiable flight-tested databases that verify that roll-yaw-pitch-TVC provides the highest payoffs at the weakest domains of conventional flight control , i.e., at low (or zero) speeds, conventionally-uncontrolled spins, very-short runways, and during PST, Rapid-Nose-Pointing and Shooting [RaNPAS] maneuvers.

Partial [or "mixed"] Jet Control (PJC) is employed when ailerons, elevators, rudders, canards, etc., are used in conjunction with TVC. Hence, any upgrading program of extant fighter aircraft, by adding TVC, is, a-priori, limited to PJC. TVC-effectiveness must then be quantified vis-a-vis the highest possible PVA standard data, roughly as an analogy to the quantification of the celebrated 'ideal cycle' as a reference standard in thermodynamic processes.

Such pure and mixed PST-TV-prototypes have been successfully flight tested by this laboratory since 1987 [1-4]. The yaw-pitch and roll-yaw-pitch TV-nozzles employed are scaled-down versions of novel yaw-pitch and roll-yaw-pitch TV-nozzles developed by this laboratory since 1983, using a Marbore jet engine installed inside an altitude test facility. Multi-axis-jet-deflections have been employed in both the laboratory and flight tests to orient the jet efflux in the required yaw, pitch, roll, and forward thrust coordinates of the engine/vehicle. Using these novel vehicles it was partially-verified that there is no danger to enter into spin situations and rapidly and completely recover by introducing strong TV-moments, thereby breaking away from the unpropitious external-flow regime [4, 5].

Proposed Mathematical Phenomenology

Using the 'ideal reference' concepts, a general mathematical phenomenology is constructed next for pure roll-yaw-pitch PST-TV. Dimensional and dimensionless terms which represent

the most general options of pure PST-TV are first presented in terms of their inherent meanings and limitations. These formulations define the basic variables of pure and mixed TV-aircraft and quantify the parameters required to maximize PST-TV control moments and rates during "g-onsets/whippings/reversals" and new combat maneuvers. The phenomenology includes a new formulation/methodology for flying powered, dynamically-scaled, pure or mixed TV-prototypes during standard agility comparison maneuvers [SACOM]. Basic rules affecting the studies of certain time derivatives that infer far-reaching consequenses for super-agile fighters are finally analyzed.

The phenomenology is characterized by the bold assumption that to describe the complex aerodynamics and rigid-body rotations/translations of advanced PST-TV-aircraft one may still use the conventional, first-order partial derivatives of flight mechanics as an approximation. This is certainly not precise in the deep PST domain, nor rigorous for the complex rigid-body dynamics characterizing certain PST-TV-maneuvers performed and monitored recently [4]. Hence, the phenomenology is general only as a pseudo-unified framework for treating all theoretical TV terms together with the main conventional terms in a linearly-superimposed formulation. Consequently, the proposed unified formulation of the 6-degree-of-freedom equations of motion, with the yet unspecified thrust-vectoring terms, reads:

$$\dot{\alpha} = q + \{ -[\bar{q}sC_x/MV-(g/V)\sin\theta + r\sin\beta]\sin\alpha + [\bar{q}sC_z/MV + (g/V)\cos\theta\cos\phi - p\sin\beta]\cos\alpha\}\sec\beta \quad [1]$$

$$\dot{\beta} = -\{[\bar{q}sC_x/MV - (g/V)\sin\theta]\sin\beta + r\}\cos\alpha + [\bar{q}sC_y/MV + (g/V)\cos\theta\sin\phi]\cos\beta - \{[\bar{q}sC_z/MV + (g/V)\cos\theta\cos\phi]\sin\beta - p\}\sin\alpha \quad [2]$$

$$\dot{p} = \{-[(I_z - I_y)/I_x + I_{xz}{}^2/I_xI_z]qr + [1-(I_y - I_x)/I_z]I_{xz}pq/I_x + \bar{q}sb/I_x[C_l + I_{xz}C_n/I_z]\}/[1 - I_{xz}{}^2/I_xI_z] \quad [3]$$

$$\dot{q} = \bar{q}scC_m/I_y + [(I_z - I_x)/I_y]pr + I_{xz}(r^2 - p^2)/I_y \quad [4]$$

$$\dot{r} = \{[I_{xz}{}^2/I_xI_y - (I_z - I_x)/I_z]pq$$

$$- [1 + (I_z - I_y)/I_x][(I_{xz}/I_z)\, qr + (\bar{q}sb/I_z)][(I_{xz}/I_x)C_l$$

$$+ C_n]]/[1 - I_{xz}^2/I_x I_z] \qquad [5]$$

$$\dot{V}/V = [\bar{q}sC_x/MV - (g/V)\sin\theta]\cos\alpha\cos\beta$$

$$+ [\bar{q}sC_y/MV + (g/V)\cos\theta\sin\phi]\sin\beta$$

$$+ [\bar{q}sC_z/MV + (g/V)\cos\theta\cos\phi]\sin\alpha\cos\beta \qquad [6]$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \qquad [7]$$

$$\dot{\phi} = p + r\cos\phi\tan\theta + q\sin\phi\tan\theta \qquad [8]$$

$$\dot{\psi} = q\sin\phi\sec\theta + r\cos\phi\sec\theta \qquad [9]$$

Unlike conventional-control variables, the TVC-variables hardly vary with external aerodynamic parameters, such as $\bar{q}s$, α, and β. Hence, the following division of, say, T_x by $\bar{q}s$ is only a matter of keeping a unified formulation with dimensionless conventional phenomenology. Yet, as enumerated below, the TV-variables vary with altitude, Mach number, engine throttle and thrust-vectoring angles. Now, each of the following equations contains at least one TV-variable, viz;

$$C_x = C_L(\alpha, \delta_e)\sin\alpha - C_D(\alpha, \delta_e)\cos\alpha + T_x/\bar{q}s \qquad [10]$$

$$C_Y = C_Y(\alpha, |\beta|, \delta_e) + C_{Y\delta_a}(\alpha)\delta_a + C_{Y\delta_r}(\alpha)\delta_r + [b/2V][C_Y(\alpha)\, r$$

$$+ C_{Yp}(\alpha)\, p] + C_{Y\beta^*}(\alpha, \beta) + C_{Y\delta_{\Delta e}}(\alpha, \delta_e)\delta_{\Delta e} + T_y/\bar{q}s \qquad [11]$$

$$C_z = -[C_L(\alpha, \delta_e)\cos\alpha + C_D(\alpha, \delta_e)\sin\alpha] + C_{[zSC]}\delta_{TV} + T_y/\bar{q}s \qquad [12]$$

$$C_l = C_{l\beta}(\alpha, |\beta|)\beta + C_{l\delta_a}(\alpha, \delta_e)\delta_a + C_{l\delta_r}(\alpha, |\delta_r|)\delta_r + (b/2V)[C_{lp}(\alpha)\, p$$

$$+ C_{lr}(\alpha)\, r] + C_{l\delta_{\Delta e}}(\alpha, \delta_e)\delta_{\Delta e} + \Delta C_{l\beta}(\alpha, \beta) + C_{lTV}\delta_{TV} \qquad [13]$$

$$C_m = C_{m_0}(\alpha, \delta_e) + [c/2V]C_{mq}(\alpha)q + T[\Delta Z_{offset}]/\bar{q}sc + C_{mSC}\delta_{TV} + C_{mTV}\delta_{TV}$$

$$[14]$$

$$C_n = C_{n\beta}(\alpha, \beta, \delta_e)\beta + C_{n\delta_a}(\alpha)\delta_a + C_{n\beta}(\alpha, \beta)$$

$$+ C_{n\delta_r}(\alpha, \beta, \delta_r, \delta_e)\delta_r + [c/2V][C_{np}(\alpha)p + C_{nr}(\alpha)r]$$

$$+ C_{n\delta_{\Delta e}}(\alpha, \delta_{\Delta e})\delta_{\Delta e} + \Delta C_{n\beta}(\alpha, \beta)$$

$$+ C_{n\beta^*}(\alpha, \beta) + C_{nTV}\delta_{TV} \qquad [15]$$

$$T_x = C_{fg}\,[\delta_v,\ \delta_y,\ NPR]\ T_i\,[\dot{M},\ T]\cos\delta_v\cos\delta_y \quad [16]$$

$$T_v = C_{fg}\,[\delta_v,\ \delta_y,\ NPR]\ T_i\,[\dot{M},\ T]\sin\delta_v\cos\delta_y - T_z \quad [17]$$

$$T_y = C_{fg}\,[\delta_v,\ \delta_y,\ NPR]\ T_i\,[\dot{M},\ T]\cos\delta_v\sin\delta_y \quad [18]$$

This set of 18 equations completes our unified formulation for conventional, mixed, or pure TV flight tests. The equations are written for a body-axis set of coordinates.

Definitions of New PST-TV Terms

Only linear expansions of moments and forces have been employed, including the unspecified, scalar "**TV**" notation in eqs. 12-14. Physico-aerodynamic fundamentals of the TV-induced supercirculation terms in eqs. 12-14 [marked here by a sub-**SC** notation], have been reviewed elsewhere and typical contributions to lift or C_Z, C_m and C_D are depicted in **Fig. 2**. The figure distinguishes between supercirculation and direct pitch-TV-additions [via 12, 14 and 17] to lift, moments, etc.

The δ_v and δ_y angles in eqs. 16-18 are valid only for effective jet-deflections. For roll-TVC one needs two TV-nozzles [Cf., e.g., **Figs. 3, 4**] and vector the δ_v-jets in opposite directions. The roll arm is **Y**. For steady-state and transient pure sideslips maneuvers [PSM], one must vector the δ_y-jets in different directions [Cf. the Appendix]. The PSM-arm is **D** or **D***, as explained in the Appendix. The δ_v and δ_y angles for each nozzle must be measured with the nozzle(s) and inlet installed on a proper jet-engine and should not be confused with the readily measured geometric angles of pitch-flaps/yaw-vanes inside 2D-CD TV-nozzles, of external TV-paddles, or of divergent flaps inside axi-TV nozzle(s). As stressed below, only the former provide the fastest possible responses to flight-control inputs.

The T_x, T_v, and T_y terms in 10 to 12 denote direct effective TV forces in the **x, z** and **y** [body-axis] directions, respectively, as defined by 16-18. The yaw and pitch moments also depend on **Y** and **D***, respectively. No constraints are placed on maximum effective TV-angles. However, as stressed below, such an angle-limit characterizes TVC based on axisymmetric TV-nozzles. For fixed throttle, Mach number and altitude, C_{fg} varies only with TV-angles [**Fig. 7**]. Yet, it also depends on nozzle air-leakage/cooling losses and on inlet

design. T_j varies with air mass flow rate through the nozzle, $\dot{M}$, and with exit gas-velocity. Provided the nozzle's exit/throat area ratio is properly controlled [1], $\dot{M}$ remains invariant during TV [Cf., e.g., **Fig. 8**]. Thus, for PST-TV-maneuvers with fixed [military] throttle (as is frequently recommended, [1]), at approximately constant Mach and altitude, 17-18 reduce to

$$T_x = C_{fg}\,[\delta_v, \delta_y]\, T_j \cos\delta_v \cos\delta_y \qquad [16a]$$

$$T_v = C_{fg}\,[\delta_v, \delta_y]\, T_j \sin\delta_v \cos\delta_y \qquad [17a]$$

$$T_y = C_{fg}\,[\delta_v, \delta_y]\, T_j \cos\delta_v \sin\delta_y \qquad [18a]$$

C_{mTV} denotes the dimensionless pitching moment per radian generated by effective jet deflection in the pitch coordinates, while C_{lTV} denotes the dimensionless roll-TV-moment per radian due to differential jet-deflection in [split-type] single or S-type twin-engine nozzle(s) [**Fig. 3** and **Fig. 4**]. The C_m-equation contains two terms associated with pitch TV: $C_{mSC}\delta_{TV}$ and $C_{mTV}\delta_{TV}$. The first is directly affected by $\bar{q}s$, while this effect on $C_{mTV}\delta_{TV}$ is different and much less pronounced, especially for PST-vectorable engine inlets which resist distortion effects at compressor inlet at high AoA and sideslip angles at low speeds [1]. Hence, except the 'supercirculation' terms in 12-14, the other TV terms are treated irrespective of $\bar{q}s$. [E.g., during pure TVC, the dynamic pressure term which multiplies C_m in 4 is canceled out by using proper units such as those in 12]. However, only high-aspect-ratio TV-nozzles that are well-integrated with wing-trailing-edges [**Figs. 3 and 4**] increase supercirculation-generated lift during jet-down-deflections, thereby slightly-expanding the flight envelopes depicted in **Fig. 1**, but significantly decrease the approach speed in landing.

Restrictions and Approximations

There are two types of coupling: Kinematic and aerodynamic. The coupling terms cannot be neglected in the analysis of PST-TV flight, unless some bold simplifying approximations are made to generate meaningful/measurable/repeatable PST-TV-SACOMs [3, 4 and below]. Mach number effects enter the TV-phenomenology through their increase of Nozzle Pressure Ratio

[NPR] and its resulting effects on C_{fg} metrics [**Fig. 7**]. The C_{fg} metrics are functions of engine/aircraft/performance parameters such as inlet configuration/control-mode, Mach number, altitude , AoA, sidslip angle, throttle, and nozzle configuration and control modes.

Only low-speed SACOMs are analyzed next, i.e., for $M < 0.4 - 0.6$ [Cf. **Fig. 1**]. Hence, the PST-TV-maneuvers are assumed to be conducted in the incompressible flow regime. Various other effects have been neglected in this model. For instance, asymmetric/inertia effects due to fuel-distribution/sloshing, elasticity/relaxation phenomena and air-turbulence noise [Cf. eq. 19 below]. Moreover, during very rapid rigid-body-dynamics-controlled "flip-up/down" or "rotational-whippings" of the nose/bottom of the aircraft, the induced inertia/gyroscopic effects play much higher roles than with relatively sluggish current conventional fighter aircraft [4]. To maximize the highly required PST-TV-roll moment, and rates, the length of the TV-rolling arm, **Y**, must be maximized/optimized. Similar conclusions apply to **D** and $\mathbf{D^*}$ during pitch and PSM [Cf. the Appendix].

Formulation of Net Super-Agility and Dynamic Scaling

With the rapid advance of new technologies, engagement times get shorter, and the minimization of inherent delay-times of TV-nozzles, pilot, and TVC-IFPC-hardware become more critical to combat effectiveness. Hence, to simulate TV-controllability by flying powered scaled models, we define Aircraft Gross Agility **[AGA]** as

AGA = MGA [DSF] F_1[Turb.-MLEM] F_2[PDT/FDT] F_3[A-IFPC]/[M-IFPC] , [19]

where **MGA** is scaled Model Gross Agility, **DSF** the Dynamic Scale Factors, to be defined below, **F_1[Turb.-MLEM]** the functions of 'Turbulence Noise and Maximum Likelihood Estimation Method' [4], **F_2[PDT/FDT]** the ratio of pilot to flyer delay times during actual, in-flight SACOM [4], and **F_3[A-IFPC]/[M-IFPC]** the control functions relating aircraft Integrated Flight Propulsion Control [IFPC], to model-IFPC [1, 4]. Without stating it, 19 assumes that each vehicle is characterized by a hidden, <u>bona fide</u>, net agility - a basic combat-technological quality that the propulsion/airframe designer and the theoretician both want to uncover and continuously maximize.

To proceed one assumes that differences in aerodynamic effects between model and

full-scale aircraft are of 'second-order' in comparison with moments-of-inertia-related angular velocities & accelerations. This approximation is justified especially for high **Re No.** ranges and for a strict proportional size-shape similarity between the full-scale aircraft and scaled models. We therefore write:

$$[dx_i]_A/[dx_i]_M = L \quad [i = 1, 2, 3, \text{ or } x, y, z] ; \quad r_A = L\, r_M$$

$$W_M = Mg \cong g\bar{\rho}_M \int_{V_M} [dx_i]_M = g\bar{\rho}_M L^{-3} \int_{V_A} [dx_i]_A = W_A\, L^{-3}\, [\bar{\rho}_M/\bar{\rho}_A] \qquad [20]$$

$$I_M = \int_{M_M} r_M^2 dM_M \cong \bar{\rho}_M \int_{V_A} L^{-2}\, r_A^2 L^{-3}\, [dx_i]_A =$$

$$= \bar{\rho}_M L^{-5} \int_{V_A} r_A^2\, [dx_i]_A \qquad = I_A\, [\bar{\rho}_M/\bar{\rho}_A]\, L^{-5} \qquad [21]$$

$$\text{where} \int_{V_A} r_A^2\, [dx_i]_A \cong I_A/\bar{\rho}_A \qquad [22]$$

and **M** is mass, **W** weight, **r** radius, and the subscripts **M** and **A** refer to model and full-scale aircraft, respectively. $\bar{\rho}$ is the average density, **L** the linear-scale-factor, and **I, x, y** and **z** are the moment-of-inertia components and coordinates as defined in the previous equations.

Hence, for **MGA**-Angular-Reversal-Rates[ARR], namely for pitch, roll and yaw rate reversals, we write

AGA[ARR] =

$$= \mathbf{MGA[ARR][L]^{-0.5} F_1 [Turb.\text{-}MLEM]\; F_2 [PDT/FDT] F_3 [A\text{-}IFPC]/[M\text{-}IFPC]} =$$

$$\cong \mathbf{MGA[ARR][L]^{-0.5}} \qquad [23]$$

where, as a first iteration, the functionals $F_1 F_2\ F_3$ are approximated by unity, aircraft performance angles remain scale invariants and agility time is compressed by the factor $[L]^{-0.5}$.

The maximum [gross] pitch rate observed so-far with our 1/7-scale flying PST-TV-F-15 model is around 200 deg/s. By 23, it is around $[200][7]^{-0.5}$, i.e., around 3 times the current maximum 'corner' turn-rate of conventional F-15s. The errors involved in using such dynamic scale factors do not depend on any assumption related to the Model or full-scale

Reynolds or Froude numbers. Eqs. 20, 21 and 23 are therefore based on simple physical laws, irrespective of any boundary-layer assumptions. Laboratory and flight-testing verifications of these equations are available elsewhere [4] and provide repeated verifications of our methodology. Other dimensionless scaling-up numbers are enumerated elsewere [1].

Approximate Phenomenology for Comparative Studies

For a flip-up/down, cobra-type, pitch-only, 'pseudo-horizontal', PST-TV-SACOM [4], performed with PVA, or with frozen conventional control surfaces, the vehicle is very rapidly whipping the air: Up-and-down in positive and down-and-up in negative 'Cobra-type' maneuvers **[Fig. 8]**. This rapid bottom/nose-pointing, or 'rotational-whipping/onset' capability, may keep the vehicle's flight-path approximately horizontal for a short duration **[Figs. 9, 10]**. Depending on its **T/W**, stability margin, IFPC/Flyer-delay-times, Mach number, altitude and SACOM duration, the vehicle may gain some altitude prior to reversing this trend during a positive 'Cobra' maneuver, while the flight-path is consistently downward during a negative 'Cobra' maneuver. However, the low ratio of altitude change to the horizontal distance covered during very rapid maneuvers allows one to assume that, as a SACOM-approximation, the flight path remains at a "pseudo-constant-altitude". Under these conditions, $\phi, \beta, \dot{p}, p, \dot{r}, r, \dot{\beta}, \delta_e, \delta_a, \delta_r, \delta_{\Delta e}, C_l, C_n$ and C_Y vanish, while $\theta = \alpha$ and $\dot{\alpha} \cong q$. Moreover, the supercirculation term can be neglected for the low-aspect-ratio TV-nozzles of our early PST-TV F-15 flying models. This conclusion is due to the small surface area affected by such nozzles [**Figs. 3, 4**]. The term $T[\Delta Z_{offset}]$ vanishes when the thrust acts through **CG**, as is the case with all our scaled TV-vehicles. Under these conditions, the flyer command is a pure δ_v input, for which

$$C_x = C_L(\alpha) \sin \alpha - C_D(\alpha) \cos \alpha + T_x/\bar{q}s \qquad [24]$$

$$C_Y = 0 \qquad [25]$$

$$C_z = -[C_L(\alpha) \cos \alpha + C_D(\alpha) \sin \alpha] + T_V/\bar{q}s \qquad [26]$$

$$C_l = 0 \qquad [27]$$

$$C_m = C_{mo}(\alpha) + C_{mTV}\delta_v \qquad [28]$$

$$C_n = 0 \qquad [29]$$

$$Mg = \bar{q}s[\, C_x \sin \alpha - C_z \cos \alpha] \qquad [30]$$

$$\dot{q}I_y = \bar{q}sc[C_{mo}(\alpha) + C_{mTV}\,\delta_v] \qquad [31]$$

$$M\dot{V} = \bar{q}s[C_x \sin \alpha + C_z \cos \alpha] \qquad [32]$$

For this SACOM 16a to 18a reduce to

$$T_x = C_{fg}\,[\delta_v]\, T_i \cos \delta_v \qquad [33]$$

$$T_v = C_{fg}\,[\delta_v]\, T_i \sin \delta_v \qquad [34]$$

$$T_y = 0 \qquad [35]$$

Eqs. 24 to 35 are employed to generate a practical SACOM, viz., from level flight initial conditions, to level flight end conditions. It provides integral 'time-to-target-and-recover' data, maximum q rates [reached usually beyond midway up and down the flip angle], and maximum 'g-onsets on the pilot' ~~at~~ beyond the reversal point [see eq. 26a below]. The corrected and scaled g-onsets, especially the negative ones, quantify the most critical pilot tolerances in a repeatable methodology, while, simultaneously providing TVC-designers with the best yardstick for maximization of TVC effectiveness vs unpropitious domains dictated by pilot tolerances [see below].

Numerical and analytical solutions of this set [with particular initial and boundary conditions] can be investigated while working back and forth between theory and well-controlled SACOM flight tests. The tests can verify the variation of q, $\dot{q}$ and $\ddot{q}$ with the time-variations and range of the δ_v command at different true air-speeds, using eq. 31. (For this purpose, we instrument the dynamically-scaled models with 3 gyros, 3 accelerometers and α, β and V probes. During the SACOM each variable is recorded 20 times per second on an anboard computer. A synchronized ground computer simultaneously records 40 times per second the TV-commands, and, whenever used in the comparisons, the conventional-control commands. The computers and associated software/calibrations/data-bases have all been developed towards this aim. Repeatability may increase via perpendicular-to-the-wind-direction-SACOMs [4])

Mathematical Simplifications for 90-deg. AoA Reversals

This pure-TV-SACOM variant is defined by TV-commands which reverse the jet direction at exactly positive or negative 90 deg AoA, namely, when the aerodynamic 'lift' vanishes [Cf. **Figs. 9** and **10**]. This reduces 24 to 35 to:

$$C_x = [C_{fg} [\delta_v] T_j \cos \delta_v]/\bar{q}s \qquad [36]$$

$$C_z = [-C_D(90) + C_{fg} [\delta_v] T_j \sin \delta_v]/\bar{q}s \qquad [37]$$

$$C_m = C_{mo}(90) + C_{mTV} \delta_v \qquad [38]$$

$$M"g" = C_{fg} [\delta_v] T_j \cos \delta_v \qquad [39]$$

$$\dot{q}I_y = D * C_{fg} [\delta_v] T_j \sin \delta_v \qquad [40]$$

$$M\dot{V} = C_{fg} [\delta_v] T_j \cos \delta_v = T_x \qquad [41]$$

Thus, for a readily measurable SACOM

$$\dot{q} \propto C_{fg}[\delta_v] \sin \delta_v \qquad [40a]$$

provided $C_{fg}[\delta_v]$ is known from the jet-propulsion tests/calibrations [Cf., e.g., **Fig. 7**]. Analytical integrations and differentiations of 40 and 41 with various IFPC/flyer delay times are readily derived for Dirac-type TV-time-commands. These commands maximize the required PST-TVC-power. However, turbulence noise, and minor flight variations from these approximations must simultaneously be measured and the results corrected accordingly [4].

Forbidden Human Space-Time Domains

Situating an hypotethical pilot's head at CR – the 'center of rotation' [where there are no 'centrifugal' and tangential accelerations during rapid 'pure' pitch-up/down 'cobra' whippings], the normal acceleration on his head is roughly approximated by

$$G_z = \{\bar{q}s[C_L(\alpha) \cos \alpha + C_D(\alpha) \sin \alpha] + T_v\}/M \qquad [26a]$$

or, for the simplifying 90-deg-AoA-pitch-SACOM-reversal [when δ_v changes sign], by

$$G_z = \{\bar{q}sC_D(90) + C_{fg} [\delta_v] T_j \sin \delta_v\}/M \ . \qquad [37a]$$

G_z does not change sign when δ_v does **[Fig. 9]**. The hypothetical pilot starts sensing 'negative-g' [blood flow into brain] only when, at low speed/drag values,

$$\{-T_v + \bar{q}sC_D(90)\} < 0. \qquad [37b]$$

More generally, speed reduction due to $\bar{q}s[C_L(\alpha)\cos\alpha + C_D(\alpha)\sin\alpha]$ acts to defer crossing into 'negative-g' domains, for it introduces a compensating 'positive-g' component [blood flow from brain]. Situating the pilot ahead of CR adds positive or negative tangential pitch acceleration [**Fig. 9**], and allows simple calculations of total G_z for a realistic pilot.

Consequently : (i) - Crossing into negative-g domains depends on AoA, airspeed, pilot's distance from CR, $\dot{q}$, and the value, sign and duration of the δ_y command; (ii) - The higher the speed, the longer the delay time into negative-g domains; (iii) - Contrary to high positive G_z-loads which characterize conventional pitch-up maneuvers [upper graph in **Fig. 10**], the faster the nose-turning rates, or the shorter the 'time-to-target-recover-PST-TV-maneuver', the more effective, and safer, it becomes for a pilot situated 'close' to 'CR', viz., for both positive and negative pitch g-loads on the pilot:

[PST-TV-G_z-loads] < [Conv.-G_z-loads] [Cf. lower graphs in **Fig.10**];

(iv) - Adding tangential and 'centrifugal' accelerations on a pilot situated ahead of 'CR', does not change these general conclusions, even for the fastest measured PST-TV-flip-up/down ($M<0.35$).

(v) - Maximum pitch-agility is affected by airframe/engine structural 'g-limitations' at high subsonic speeds, and is hardly, if at all, influenced by pilot tolerances at low speeds.

To verify these conclusions, improve TVC designs, and study PST-TV agility and tactics, the maximized G_z, G_x, G_y, $\dot{q}$, $\dot{p}$, $\dot{r}$, α, β, and $\dot{V}$ envelopes are simultaneously measured by our dynamically-scaled models during very rapid pitch, roll, and sideslip SACOMs.

G_x during this SACOM includes positive [blood flow to chest] 'centrifugal' acceleration acting on the pilot from 'CR'. For this SACOM, the non-centrifugal/rotational component of G_x [when the hypothetical pilot is situated at 'CR'], is roughly approximated by

$$G_x = \{\bar{q}s[-C_L(\alpha)\sin\alpha + C_D(\alpha)\cos\alpha] - T_x\}/M \quad , \qquad [24a]$$

Similarly, the non-centrifugal/rotational portions of G_x and G_y can be measured and compared with load-approximations for **PSM** [Cf. Appendix], viz.,

$$G_x = \{-C_{fg}[\delta_y]\,T_i\cos\delta_y + \bar{q}sC_D[\alpha(0)]\}/M \qquad [45a]$$

$$G_y = \{-\bar{q}sC_y(\beta) - C_{fg}[\delta_y]\,T_i\sin\delta_y\}/M \qquad [46a]$$

The G_z, G_y, G_x pilot tolerances vary differently with the duration and rate of 'onsets'. Therefore, combined with such [a-priori known] duration/rate limitations, the measurment envelopes translate into forbidden human space-time agility domains for supermaneuvers.

These domains have not yet been fully explored. Their boundaries vary, inter alia, with the distance from the pilot's head to the so-called pseudo-instantaneous-center-of-rotation during different, rapid, supermaneuvers. Understanding these complex rigid-body translational, rotational, gyration, and gyroscopic phenomena, requires reassessment of a few, 'well-established', human-systems/aircraft/control/effectiveness concepts. Verification of such theoretical criteria by collecting well-defined 'flight-tested' data can therefore help the design of new centrifuge simulations [8,9] of human systems exposed to extreme PST-TV conditions, and, consequently, to establish the optimal location of the pilot's seat/head in super-agile fighters.

Radius of Gyration

The inertia tensor, I_{ij} (i,j = 1, 2, 3 or x, y, z), may be divided into an inertial tensor relative to the center of mass of the aircraft, and an inertia tensor relative to another point of reference. Hence, the quantities associated with it - principal axes, principal moments, etc. - are relative to a particular point of reference.

If the reference point is shifted from the center of mass of the aircraft to another point, as is required for improved understanding of pilot-induced rotational-agility limitations, these quantities change accordingly. The combined translational-rotational dynamics of, say, pure-pitch SACOMs, may similarly be split into two separate formulations, one purely translational and the other purely rotational about a reference point. To simplify the formulations of rigid-body dynamics and flight tests of PST-TV vehicles, one may employ the radius of gyration, which is directly related to the moments of inertia. For instance, the radius of gyration around the pitch axis of the PST-TV vehicle, R_y, is defined by

$$R_y = [I_y/M]^{0.5} \qquad [42]$$

where M is the mass of the flying vehicle [7]. Flight tests conducted by this laboratory employ the radius of gyration formulation to extract improved understanding of pilot tolerances in dynamically-scaled-up, yaw-pitch, or roll-yaw-pitch-PST-TVC F-15, F-16 and

F-22 fighter aircraft upgrades.

The measurements of G_z, G_y, G_x-envelopes - and of forbidden human space-time domains for each of these upgrades, the verification of the radius of gyration methodology for the new SACOMs, and the development of mathematical approximation methods, are sponsored by the USAF/AFOSR/EOARD, U.K. The test data are employed by the Armstrong Laboratory at WPAFB, Ohio, and by the Human System Division at BAFB, Texas. Currently there is no other source for such data [8,9].

A Basic Effectivity Rule

The following criteria introduce additional forbidden space-time domains: Whenever TVC is required, the jet-rotation rates should not lag behind the maximum rotation rates extractable from advanced conventional elevators, rudders and ailerons, viz., effective TVC rates cannot lag behind the conventional ones. This is a basic effectiveness rule that can affect the future of vectored aircraft, for it forces the selection of the most-effective TV-nozzles and TVC modes for proposed new or upgraded fighter aircraft [see below].

To quantify and gauge this rule one may first re-examine the time derivatives of eqs. 31, 34, 40 and 40a. For instance, the time derivative of pitch jet-deflections, $\dot{\delta}_v$, affects the second time derivative of the pitch rate, $\ddot{q}$, and, hence, the pilot's ability to stop or reverse rapid unwanted, or wanted, turning rates/oscillations under conventional or PST conditions. This may help gauge pilot's ability to minimize offensive or defensive delay times with TVC. An example is provided next.

Effectiveness Conclusions

Most axi-TV-nozzle-delay-times result from excessive complexity and the high inertia/friction associated with a large number of moving/sliding flaps/spacers and extra rings and sliding rods-ducts **[Fig. 11]**. Moreover, the divergent flaps/spacers touch each other under maximum TV geometric deflections, thereby restricting maximum possible TV-deflection angles to about 20 degrees. In comparison, there are only a few, non-sliding, rotating/deflecting flat vanes/flaps in yaw-pitch 2D-CD nozzles, with no restrictions attached to maximum jet-deflection angles.

Both 2D and axi TV-nozzles are practically similar from the combined point of view of

engine reliability/performance, AB-duct/airframe structural reinforcement, actuators-sizes, and structure/weight/control/cost criteria required for adding TVC.

Consequently, yaw-pitch TV-2D-CD nozzles must be selected via the aforestated rule, for they provide the maximum possible jet-deflection rates. (The internal yaw vanes may be internally cooled, as, for instance, the advanced, 1st-stage, turbine stators are.)
Similar rate-factors, and the basic disadvantages of external TV-paddles, have been investigated with the X-31 and F-18.

Therefore, success criteria in air-to-air and in air-to-ground TVC, or under STOL or spin-avoidance-recovery conditions [5], include super-fast-responding yaw-pitch or yaw-pitch-roll 2D-CD TV-nozzles. Consequently, such engine nozzles are being developed in this laboratory [via contracts with PWA].

Concluding Remarks

Fundamental concepts of multifunctional, pure and mixed TV-aircraft have been defined and employed in the construction of a unified mathematical phenomenology for PST, roll-yaw-pitch TV-controlled aircraft, and as "Ideal Standards" to maximize control speed/autonomy and nose-turning-rates during PST-maneuvers. The phenomenology and the deduced mathematical approximations have helped establish:

a) - A unified methodology for integrating theory with jet-propulsion and flight tests of dynamically-scaled, powered, PST, yaw-pitch, or roll-yaw-pitch TV-prototype models.

b) - Basic PST-TV-yardsticks of merit vs proper testing methodology to assess forbidden PST-TV-space-time domains, in particular the yet unexplored domains of maximum pitch-PST-TV-agility components which are mainly affected by aircraft or engine structural 'g-limitations' at high subsonic speeds, and those which might be slightly affected by pilot tolerances at low speeds and prolonged supermaneuvers.

c) - New basic criteria and computer hardware/software/calibration-metrics to measure correct agility/propulsion/control effectiveness in the jet-laboratory and during flying standard agility comparison maneuvers with scaled PST-TV-models or with full-scale aircraft.

d) - A critical study of the origins of slow jet-deflection rates associated with axi-TVC in comparison with maximum TVC rates extractable from low-aspect-ratio, novel yaw-pitch

TV-2D-CD nozzles. Since such delay times can become critical in air-to-air and air-to-ground missions/tactics in which conventional flight control must be supplemented or replaced with TVC, the new super-fast yaw-pitch 2D-CD nozzles incorporate far-reaching consequenses for the future of super-agile fighters.

Appendix

New Pure-Sideslip-Maneuvers With Tailless Vectored Fighters

Tailless, pure, or "ideal" thrust-vectored aircraft can perform Pure Sideslip Maneuvers [PSM] with constant [steady-state] heading, or as rapid-nose-turning transients, viz., without banking. During steady-state PSM one TV-nozzle deflects the jet in the yaw direction until its vector coincides with the side-center-of-pressure, C_{py}. This causes PSM with zero yawing-rate and banking, i.e., $\dot{r}$, $\dot{p}$, $\dot{q}$ and $\dot{\beta}$ vanish, but not β. [To perform this SACOM, the non-yawing, axial thrust generated by the 2nd TV-nozzle is reduced to equal that left-over by the 1st nozzle, so as to avoid a yawing moment on the TV-aircraft, unless transient maneuvers are required.]

During maximization of transient PSM, both nozzles are yaw-deflected in the same direction, at the fastest rate to maximum specific design-limit of δ_y values. The aim of such maneuvers with tailless configurations is to acquire the target and rapidly recover with minimal energy dissipation. [A similar PST-TV acquisition dissipates considerably more energy. Hence, to acquire any target in space-time, such a PSM-yaw is <u>a-priori</u> combined with a well-calculated roll [1]]. A simplified mathematical phenomenology for assessing such advanced systems is provided next.

Consider the simplest steady-state PSM SACOM at zero AoA and zero pitch attitude with no banking and roll. During such a SACOM with pure TVC, the TV forces and moments replace the conventional ones. For a preliminary analytical assessment the conventional coupling between yaw and roll through the tail, [asymmetric-flow-over] wing, etc., are assumed negligible for tailless pure-TVC configurations [Cf., e.g., **Figs. 3 and 4**]. Under such bold approximations the α, $\dot{\alpha}$, θ, ϕ, $\dot{p}$, $\dot{q}$, q, r, δ_v, δ_e, δ_a, δ_r, $\delta_{\Delta e}$, $T[\Delta Z_{offset}]$, C_z, C_l, C_m, C_n terms vanish, and from eqs. 2, 6, 10 and 11 one obtains,

$$C_y \cos\beta = C_x \sin\beta \qquad [43]$$

$$\dot{V}/V = [\bar{q}s/MV][C_x \cos\beta + C_y \sin\beta] \qquad [44]$$

$$C_x = [C_{fg}[\delta_y]\, T_j \cos\delta_y] / \bar{q}s - C_D[\alpha(0)] \qquad [45]$$

$$C_y = C_y(\beta) + [C_{fg}[\delta_y]\, T_j \sin\delta_y]/\bar{q}s \qquad [46]$$

$$T_x = C_{fg}[\delta_y]\, T_j \cos\delta_y \qquad [47]$$

$$T_v = 0 \qquad [48]$$

$$T_y = C_{fg}[\delta_y]\, T_j \sin\delta_y \qquad [49]$$

δ_y incorporates two independent commands: one for each TV-nozzle [5]. Maximum possible steady-state-β-heading increases with **Y/D** values, while transient PSM-rates vary with **D*** and range and rate of change of δ_y. Therefore, the fastest PSM-reversal SACOM [3] is extractable with both nozzles performing the same reversal jet-deflection, starting from zero heading, and reversing when the target has been acquired.

PSM is maximized only with high-aspect-ratio, split or s-type 2D-CD nozzles of the type depicted in **Figs. 3** and **4**. Yet, low aspect-ratio, or two axisymmetric TV-nozzles can produce reasonable PSM with tailless configurations. The combat-effectiveness of the latter is, however, limited by slow δ_y and δ_v flight-control commands, inherently shorter **Y**-moment-arms, normally, higher installed C_D values and IR/RCS signatures and the absence of $C_{[zSC]}\delta_{TV}$ and $C_{mSC}\delta_{TV}$ contributions to normal forces and moments via

$$C_z = C_{[zSC]}\delta_{TV} \qquad [12a]$$

$$C_m = C_{mSC}\delta_{TV} \qquad [14a]$$

During pitch-down TVC, the $C_{[zSC]}\delta_{TV}$ and $C_{mSC}\delta_{TV}$ terms help generate the slightly-expanded flight envelope depicted in **Fig. 1**, and contribute to lower extractable approach speeds in landing.

Acknowledgements

Part of this research has been financially sponsored by the **U.S. Air Force Office of Scientific Research, EOARD, U.K.**, under (i) - Grant Number AFOSR 89-0445, and (ii)

- Special Project SPC-91-4003. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. Additional financial support has been provided by **General Dynamics, General Electric, Teledyne and Pratt and Whitney,** whose regulations prevent us from providing here additional technical details and drawings.

We wish to thank the following individuals for encouraging and trusting us with an unorthodox reserach work: **Dr. G. Keith Richey, Mr. Douglass Bowers, Maj. T. Speers, W. Lindsay, Capt. J. Wigle, Dr. W. Calarese, Col. John Tedor**, and **Dr. Daniel W. Repperger**, all from the USAF; Messrs. **Ben Koff, R. E. (Bob) Davis, Jeffery Schweitzer, and J. Cahill,** from PWA; **Mr. Eli Benstein** from Teledyne CAE; Messrs. **J. Roug, Don Dunbar and F. Ehric** from General Electric; Mr. **T. P. McAtee** from General Dynamics, and last, but not least, the late **Dr. W. B. Herbst** of MBB.

References

1 - Gal-Or, B., "Vectored Propulsion, Supermaneuverability and Robot Aircraft", Springer Verlag, N.Y.-Heidelberg, 1990, 1991.

2 - Ibid., "The Fundamental Concepts of Vectored Propulsion", [AIAA] J. Propulsion, Vol. 6, No. 6, Nov.-Dec., pp. 747-757, 1990.

3 - Ibid., "Maximizing Post-Stall, Thrust-Vectoring Agility and Control Power", [AIAA] J. Aircraft. In press. **[A copy enclosed]**

4. Ibid., "Tailless Vectored Fighters", Flight Dynamics Directorate, WPAFB, USAF/AFOSR 89-0445, July 15, 1991.

5 - Baumann, D. D., "F-15B High-Angle-of-Attack Phenomena and Spin Prediction Using Bifurcation Analysis", M.Sc. Thesis, AFIT, WPAFB, Ohio, Dec. 1989; AFIT/GAE/ENY/89D-01.

6 - Herrick, P., "Fighter Aircraft Affordability, Survivability and Effectiveness through Multi-Function Thrust-Vectoring Nozzles", AIAA-89-2815. To be published in Intern. J. Turbo and Jet Engines.

7 - Goldstein, H., 'Classical Mechanics', 2nd-ed., Addison-Wesley, Reading, Mass, 1981. pp. 202-204.

8 - Repperger, D. W., " Minimum Coriolis Algorithms For Motion Control", Armstrong Laboratory, WPAFB, Dayton, Ohio, 1990.

9 - Repperger, D. W., " A Study of Supermaneuverable Flight Trajectories Through Motion Field Simulation of a Centrifuge Simulator", ASME J. Dyn. Meas. and Control. In press.

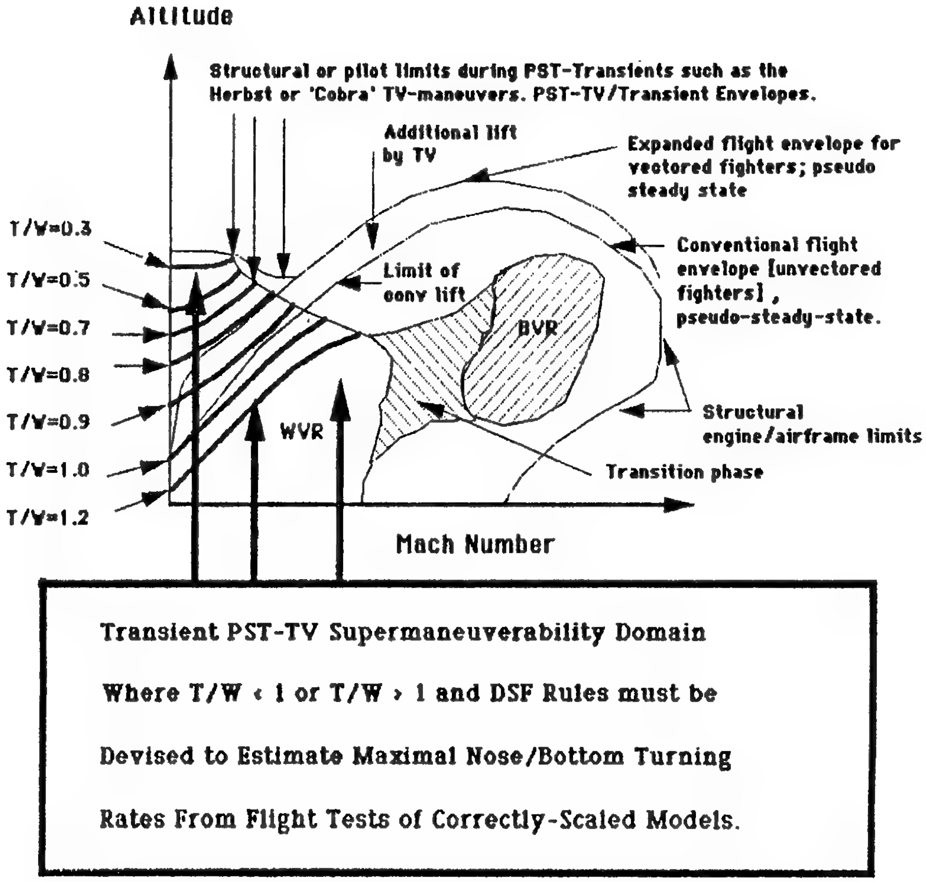


Fig. 1 : The first domain of PST-TV.
For other PST-TV-domains, including forbidden human PST-TV-domains and DSF rules, see text.

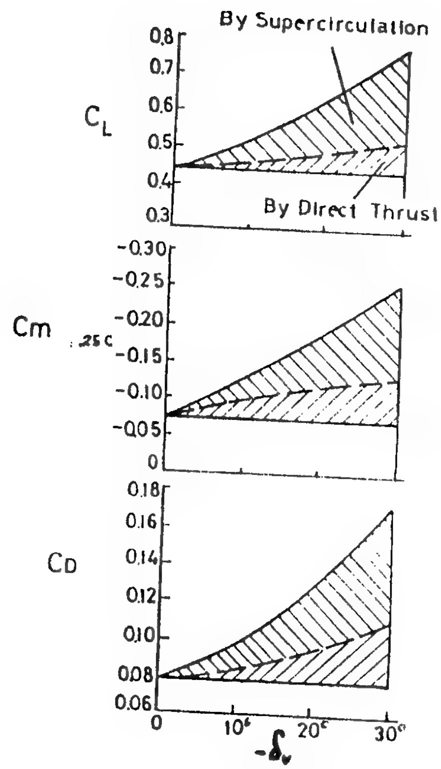


Fig. 2 : An example of Supercirculation-enhanced-lift generated by pitch-down jet-deflections on top of direct normal thrust-vectored addition to lift. TV-induced moments and drag also vary with TVC-jet-deflections in the pitch and yaw coordinates.

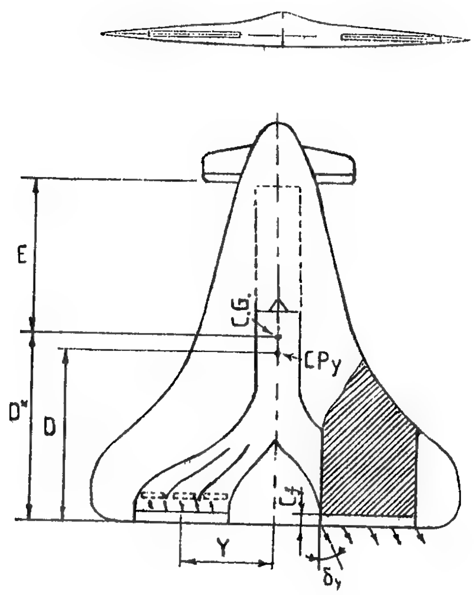


Fig. 3 : The fundamental features of Pure Vectored Aircraft (PVA)
The shaded area represents super-circulation affected wing area. The propulsion system is imbedded in the fuselage and includes roll-yaw-pitch thrust-vectoring nozzles. The canard is not an essential element of PVA. The novel unmanned vehicles flight-tested in 1987 by this Laboratory have been constructed according to these features. These PVA criteria help upgrade F-15, F-16 and F-22 fighter aircraft.

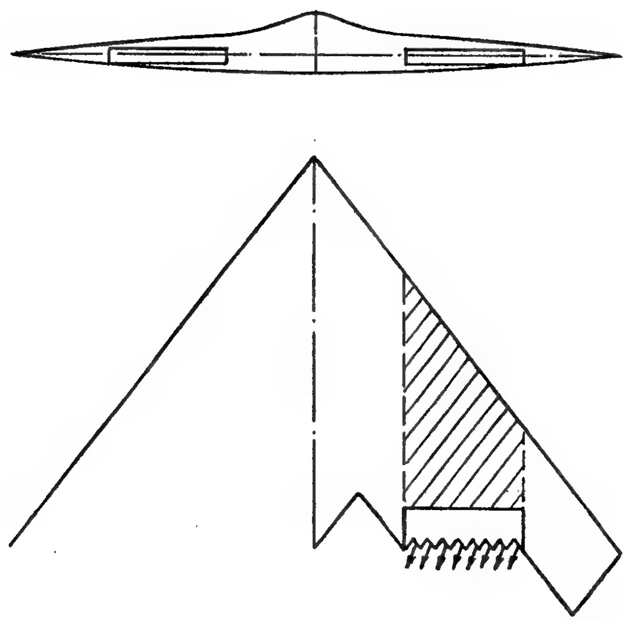


Fig. 4 : PVA with reduced signatures.
Shaded area represents supercirculation-enhanced-lift area.

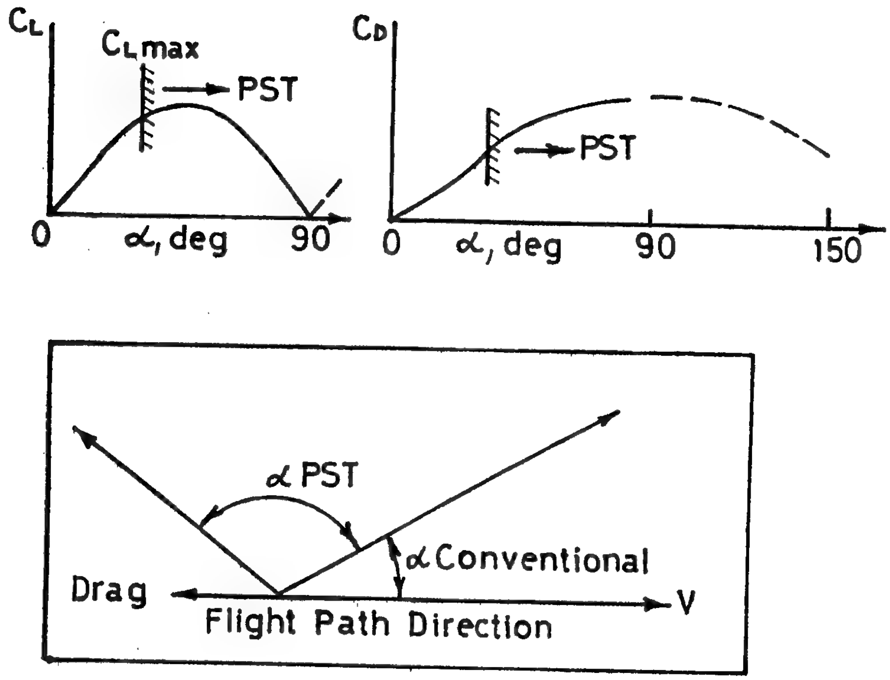


Fig. 5 : The definition of PST domain.

Note : At AoA = 90 degrees the lift vanishes, drag is maximized and roll becomes PSM.

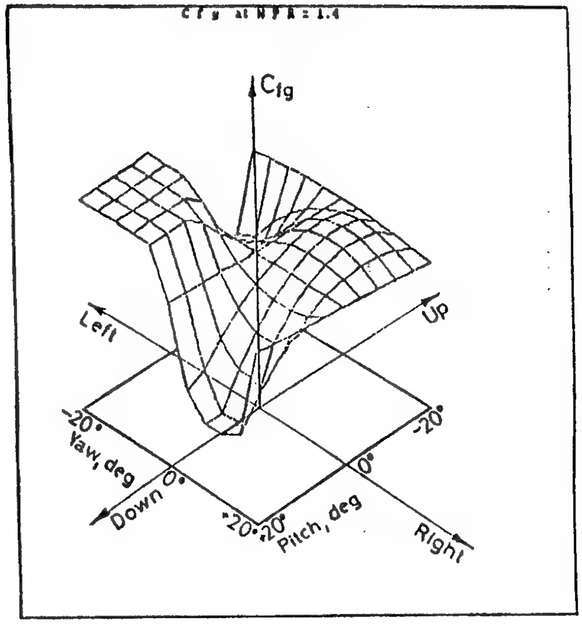


Fig. 7 : Cfg varies with yaw-pitch thrust-vectoring angles. The TVC metrics also depend on NPR. Such metrics are required for any SACOM.

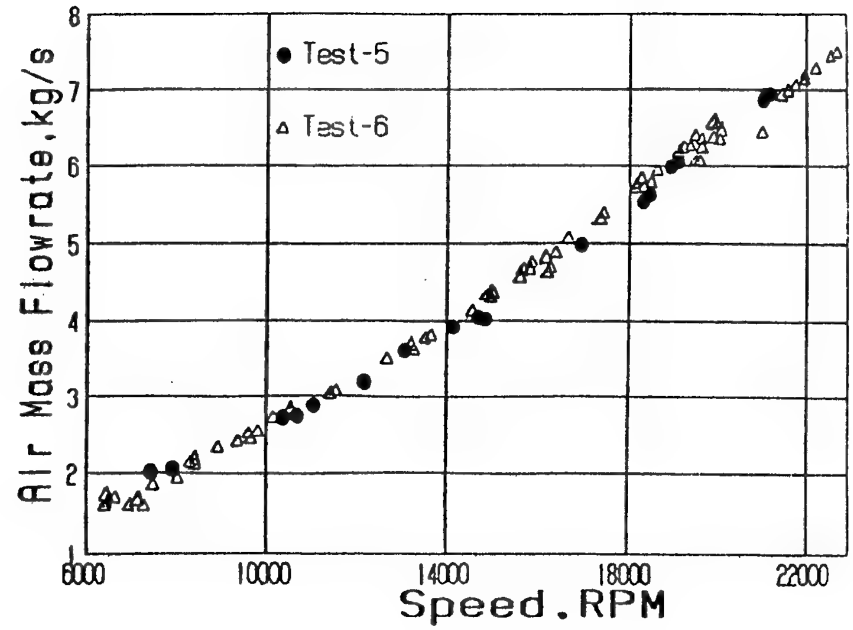


Fig. 8 : Up to 20 degrees jet deflections (test 5) do not change the mass flow rate through the engine in comparison with no jet deflection (Test 6).

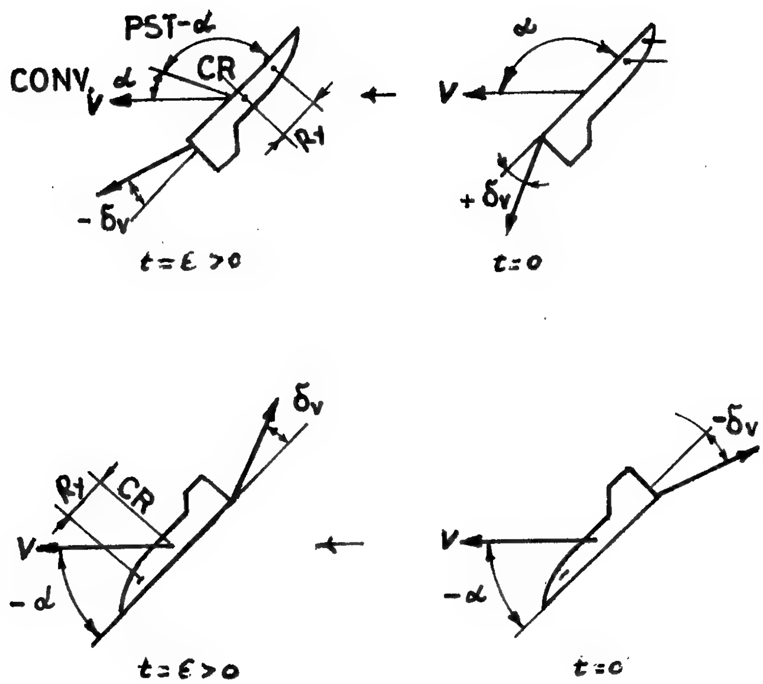


Fig. 9 : Positive and negative "Cobra" supermaneuvers. Effectiveness rule No. 1 requires TVC-rates not to lag behind rotational rates of conventional control, e.g., elevators, rudders and ailerons. Rule No.2 requires maximization of TV moments and rates at the reversal of pitch, roll and yaw supermaneuvers. Pitch TVC reversal is depicted here together with the radius of gyration. The figure represents the main features which require attention during pure-pitch SACOM. CR is the center of rotation.

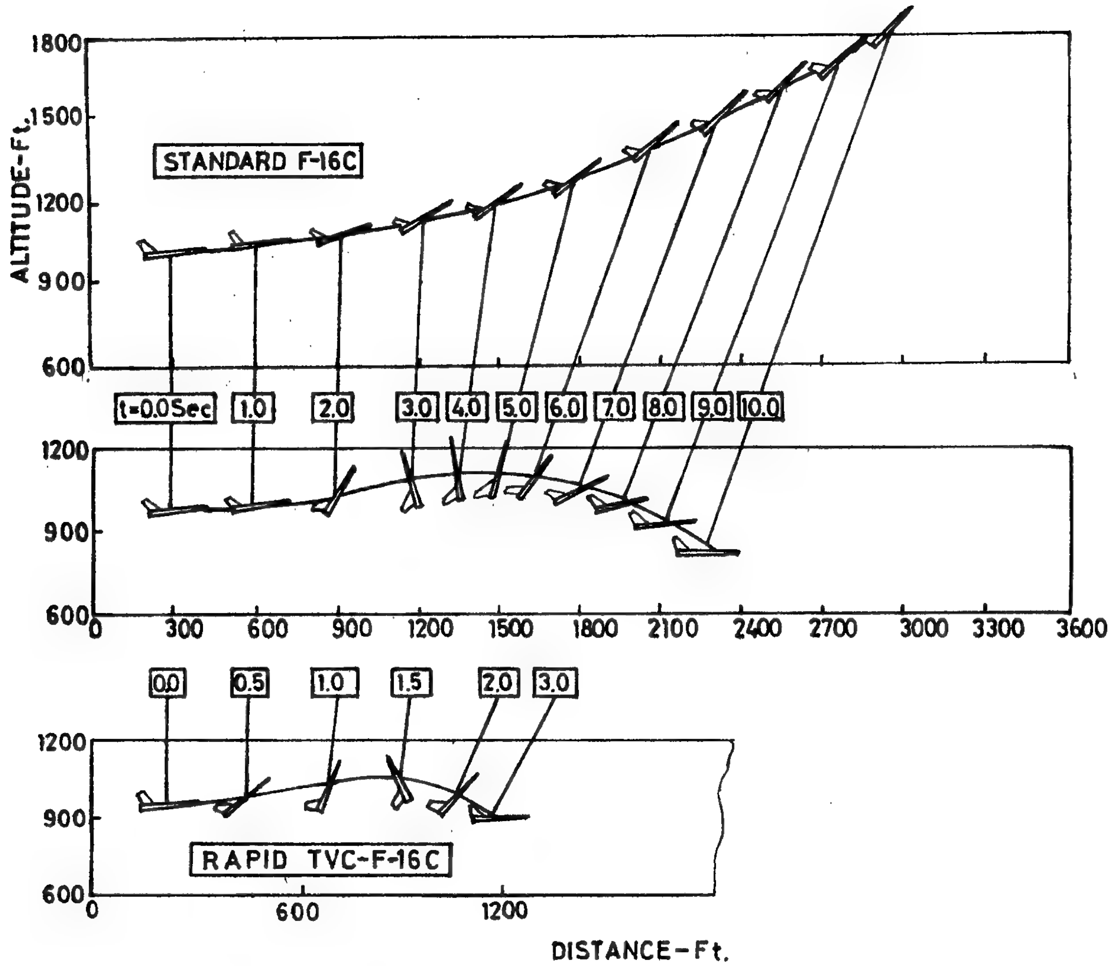


Fig. 10 : Effective TVC (lower figure) means rates which do not lag behind maximum conventional PST-rates (middle figure). Upper figure shows maximum conventional (AoA-limited) pitch-up flight control.

TV-nozzles rates must be increased from current figures (about 40 deg/sec jet-deflection rate) to beyond 400 deg/sec (cf. Fig. 11).

TVC can "aquire-target-and-recover" at minimum time, thereby minimizing missile-flight-path/time-to target and maximizing residual speed/energy.

Most important : The faster-the cobra maneuver the safer it becomes to the pilot, namely, the conventional pitch-up (upper figure) generates the highest Gz loads on the pilot. (Upper 2 graphs are based on data available in the public domain. Lower graph is based on our DSF rules and flight-tests of dynamically-scaled PST-TV-F-16 and F-15 models equipped with rapidly rotating TV-jets).

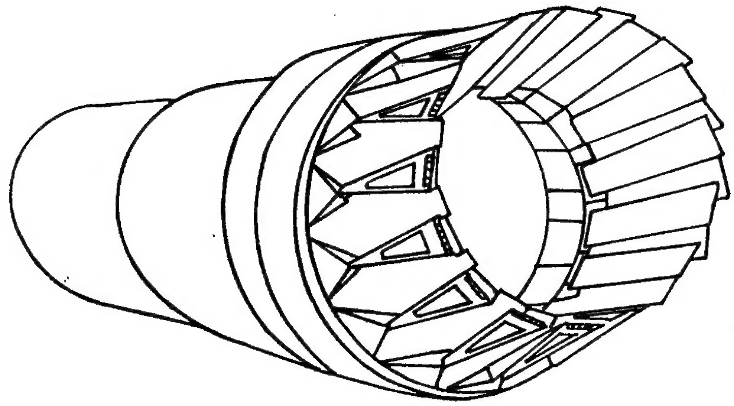


Fig. 11 : Axi-TV nozzles are characterized by a large-number of divergent flaps/spacers, and rods. Such nozzles compete with novel yaw-pitch 2D-CD TV nozzles for extracting the fastest possible TVC.

Benjamin Gal-Or **Aug. 8, 1991**

Born Aug. 8, 1933 [Israel]; B.Sc. (1959); M.Sc.(1961); D.Sc.(1964); all from the Technion - Israel Institute of Technology [TIIT]

Married [Dr. Leah Schulwolf/Gal-Or, Head, Center for Noble Metals, Israel Inst. of Metals, TIIT], 2 sons [Combat pilots],

Teaching, positions and other activities:

Assist Prof. [1964-5], The Johns Hopkins Univ ; Assoc Prof [1966-67], Univ of Pittsburgh; Assoc. Prof. [1968-71], Faculty of Aerospace, TIIT, Prof [1972-].

Founder and Head [1973 -] Turbo and Jet-Engines Laboratory, [Jet Propulsion Laboratory].
[Now contains about $ 10 million worth of R&D facilities designed and constructed via research contracts, mainly in the field of thrust-vectoring propulsion/flight-control].

Editor-in-Chief, Intern. J. of Turbo and Jet-Engines [1984 -].

Director, "Neaman Inst. of Advanced Sci. & Tech. [under development]", TIIT, [1976-8];

Chairman, ASME International Conference on Turbo and Jet Engines; Israel, July, 79

Co-chairmanship of 2 Physico-Chemico Hydrodynamics Conferences:

- Oxford University [1977];
- US National Academy of Science, Washington DC, [1978].

Member of: 1 -AIAA-Air-Breathing Comt [1975-77];

2 - New York Academy of Sciences; Sigma Xi;

3 -AIAA - Aircraft Comt. [Foreign Candidate, 1990 -]

Founder and Chairman, Israel Gas Turbine Association - "IGTA", [1979 -].

Graduate and undergraduate courses on "Technology of Jet Engines", "Jet-Engines", "Fluid-Dynamics, Heat-Transfer, and the Performance of Jet-Engines", "Design", "Thermodynamics", "Compressible Fluid Dynamics", "Viscous Hypersonic Fluid Dynamics", "Irreversible Thermodynamics"; "Astrophysics" and "Philosophy of Science" for the entire student body at TIIT.

Books:

1 - Vectored Propulsion, Supermaneuverability, and Robot Aircraft ; Springer-Verlag, N.Y. - Heidelberg, 1990, 1991.

2 - Cosmology, Physics and Philosophy, Springer-Verlag, N.Y., 1st Ed. 1981, 2nd Ed. 1983, 3rd Ed. 1987.

3 - Academe, Industry and Government, The Neaman Foundation for Advanced Science and Technology, 1979.

4 - Modern Developments in Thermodynamics. Wiley, N.Y., 1974.

5 - A Critical Review of the Foundations of Classical and Relativistic Thermodynamics, Mono Book, Baltimore, 1970

6 - Turbo and Jet Engines, Vol I, 1980, [ASME-IGTA], Vol II published by the Int. J. Turbo and Jet Engines, 1984.

Papers and Patents

91 papers in international professional journals. 3 patents [thrust vectoring].

Recent Papers.

1 - The Fundamental Concepts of Vectored Propulsion, AIAA J. Propulsion, 747 - 757, Nov.-Dec. 1990

2 - Maximizing Post-Stall, Thrust-Vectoring Agility, AIAA J. Aircraft. In press.

3 - Mathematical Phenomenology for Multifunctional Thrust-Vectored Aircraft, AIAA J. Aircraft. Accepted for publication with revision

4 - The Fundamental Concepts of Vectored Aircraft. Submitted for publication.

Aerospace Award: The Laskovitch Award and Gold Medal for best original research work in aerospace; from the New-York Academy of Sciences, 1971/1972.

D.Sc. & M.Sc. Supervisor: Supervised 10 D.Sc. and 16 M.Sc. graduates.

Current Research Projects

Recent R&D projects have been paid for by **US Air Force, General Electric, General Dynamics, Teledyne, Pratt and Whitney, Refael, DoD, Israeli Navy,** and other bodies.

- First flights of Thrust-Vectored-Controlled [TVC] unmanned vehicles [1987]. First TVC 'Cobra' and Herbst maneuvers using 9-feet, computerized/instrumented, TVC-F-15 models [1989-91].
  [Cf. Aviation Week, May 18, p. 21, 1987; 'All the World Aircraft', 1988/9 - RPVs [Israel], Flight International, 7-3 March 1990, p. 18.
- Novel yaw-pitch and roll-yaw-pitch TVC engine-nozzles/airframe systems.
- Post-stall [PST] TVC-agility definitions, mathematical phenomenology, and flight tests.
- Vectorable. Post-Stall engine-inlets: Theory and laboratory tests.

Previous Research Topics

- Supersonic and Hypersonic Plasma-Jet Guns: Mathematical Phenomenology and Applications.
- Pressurized Fluidized-Bed Combustors linked to Gas Turbines;
- Dust-Filtration systems for Helicopters and Tanks.
- Mathematical phenomenology of chemically-reacting two-phase flows and combustion systems.
- The Origin of gravitational, electromagnetic and other thermodynamic irreversibilities;
- Radiation Thermodynamics in large space systems;
- The Physics and Philosophy of Asymmetries;

List of D.Sc. and M.Sc. Graduates/Universities

1 - H. Doctor, M.Sc. [Hopkins]
2 - Tark, M.Sc. [Hopkins]
3 - Camilus, D.Sc. [Pittsburgh]
4 - Tavlarides, L., D.Sc [Pittsburgh]
5 - Waslo, S., D.Sc. [Pittsburgh]
6 - Padmanabhan, M.Sc. [Pittsburgh]
7 - Weis, D., D.Sc. [TIIT]
8 - Yaron, I., D.Sc. [TIIT]
9 - Tambour, Y., D.Sc [TIIT]
10 - Lior, D., D.Sc. [TIIT]
11 - Zehavi, E., D.Sc. [TIIT]
12 - Tal, R., D.Sc., D.Sc. [TIIT]
13 - Bar Anan, D., D.Sc. [TIIT]
14 - Maxim, M.Sc. [TIIT]
15 - Ben Nahum, M.Sc. [TIIT]
16 - Narkis, M.Sc [TIIT]
17 - Lustig, J., M.Sc. [TIIT]
18 - Tidhar, M.Sc. [TIIT]
19 - Gali, R., M.Sc. [TIIT]
20 - Cheruinik, M.Sc [TIIT]
21 - Ben-Shmuel, M.Sc. [TIIT]
22 - Wiswanatan, M.Sc [Pittsburgh]
23 - Levy, S., M.Sc. [TIIT]
24 - Shamash, M.Sc [TIIT]
25 - Michael, M.Sc. [TIIT]
26 - Barnea, M.Sc [TIIT]